Midway Design Review

Team Castle Quest

December 5, 2016

Team Castle Quest

David Lassalle
CSE

Devrim Dereli
EE

Eric Wybenga
CSE

Sarah Mangels
CSE

Electronic Board Game for the Modern Player

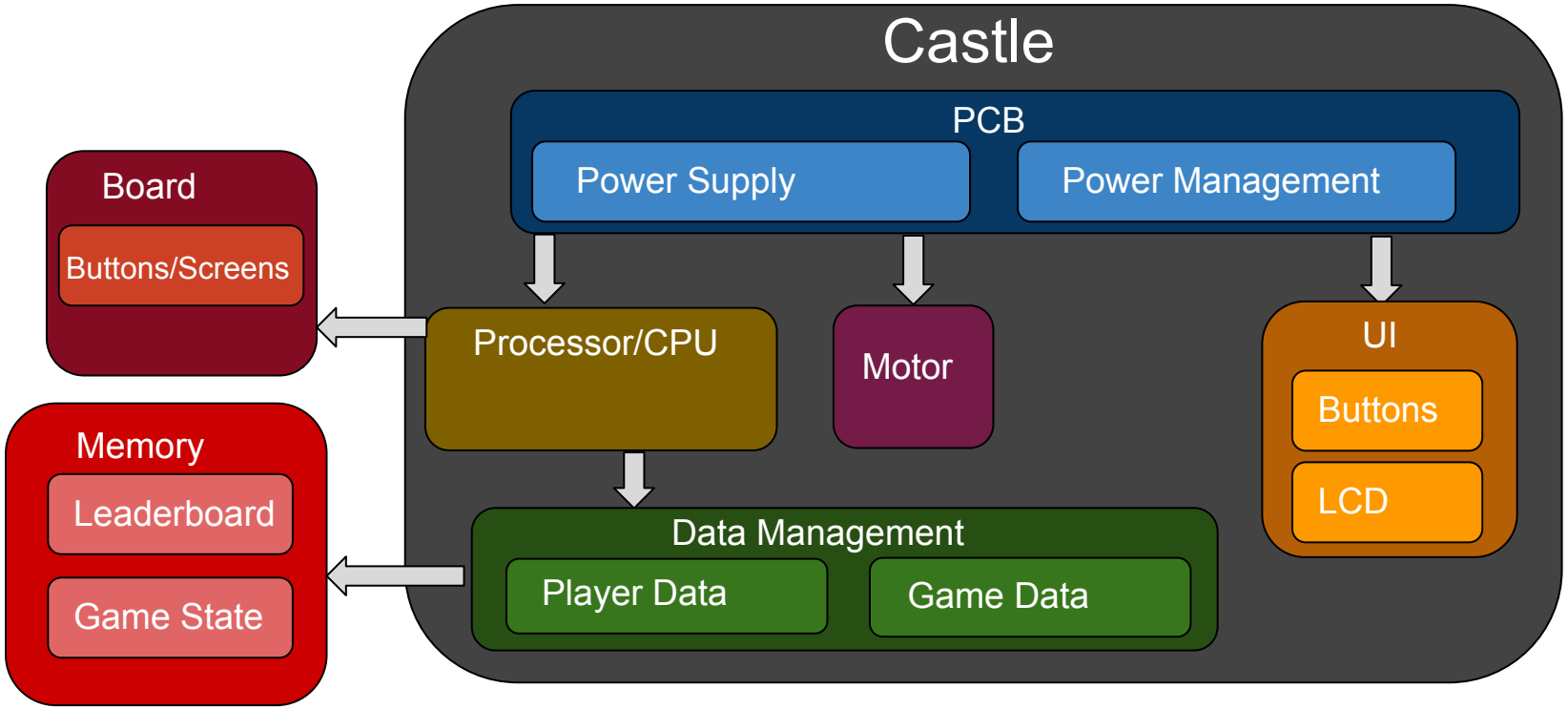
- Electronic entertainment is no longer group-centric
- Virtually multiplayer, physically singleplayer
- Market and psychological demand for group centric games, especially board games

- Castle Quest is a electronic 4-player fantasy board game centered around an electronic Castle

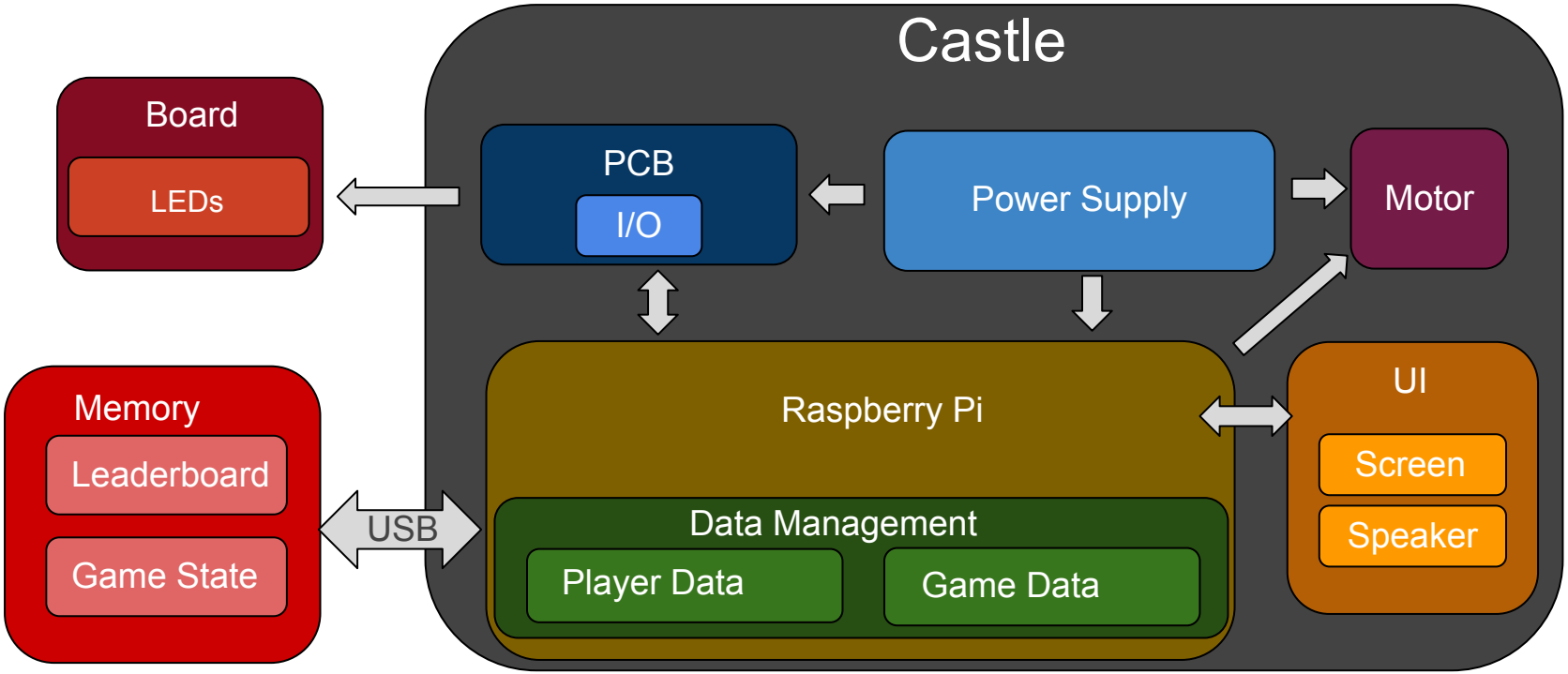
System Requirements

- Meet child safety standards of Consumer Product Safety Commission
- Game should fit into a $1ft^3$ container
- Game should weigh < 5 lbs
- 10+ hours battery life
- Durability to survive transport and repetitive construction/deconstruction
- Support for up to 4 players
- Simple, intuitive gameplay mechanics
- Inexpensive
- Fun to play

Previous Block Diagram

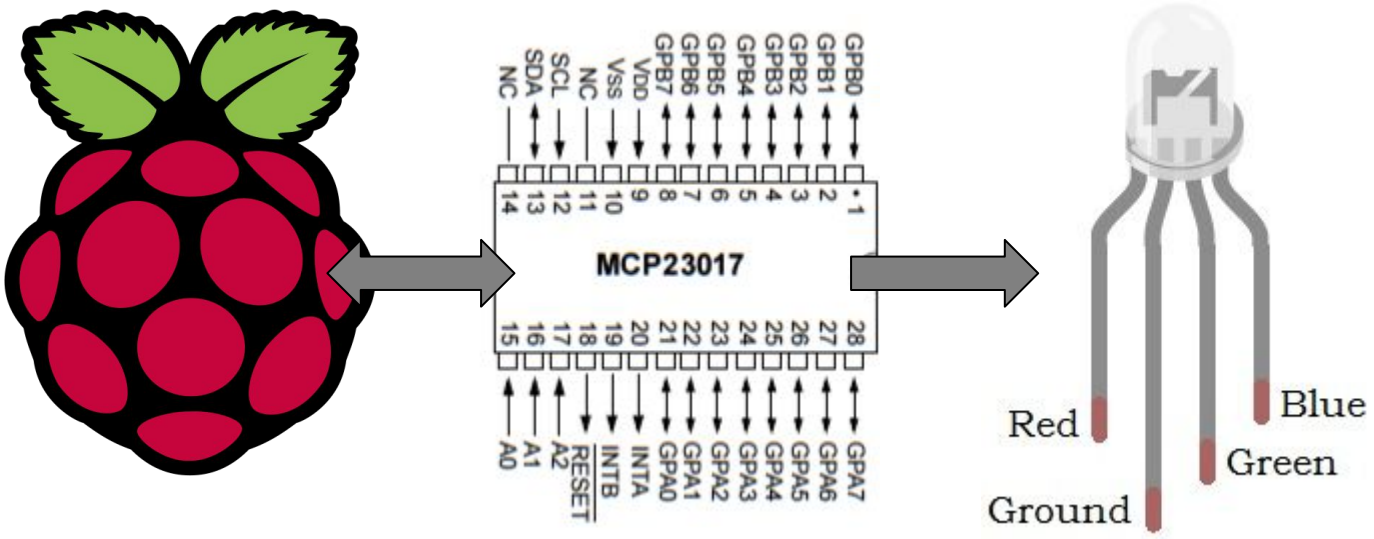


Updated Block Diagram



Design Changes

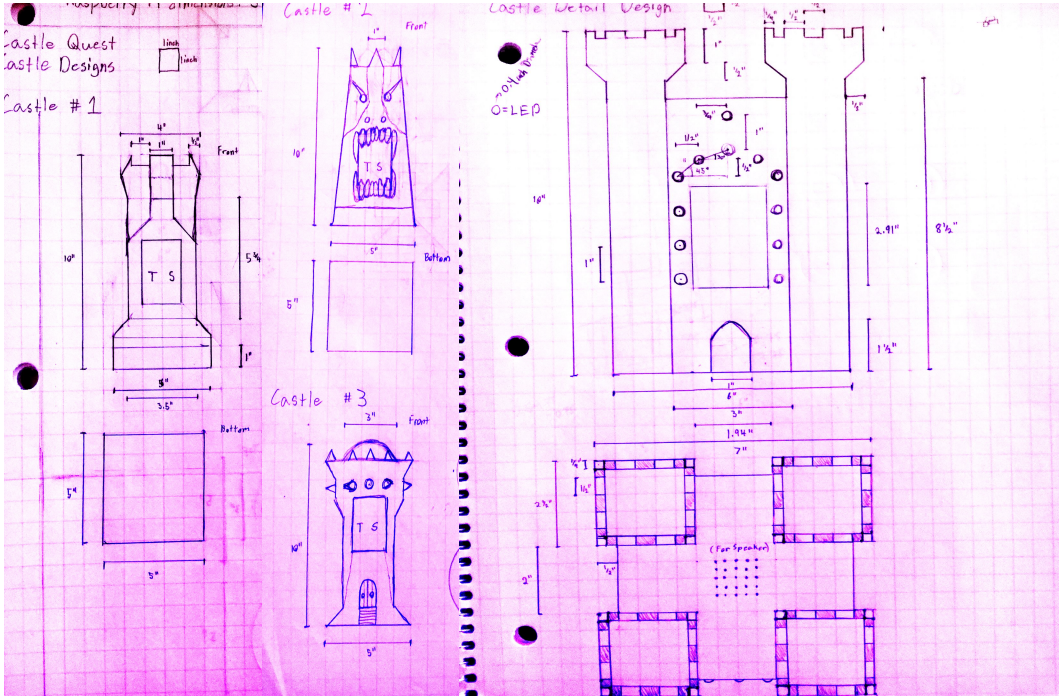
- LED position display on board
- PCB handles I/O

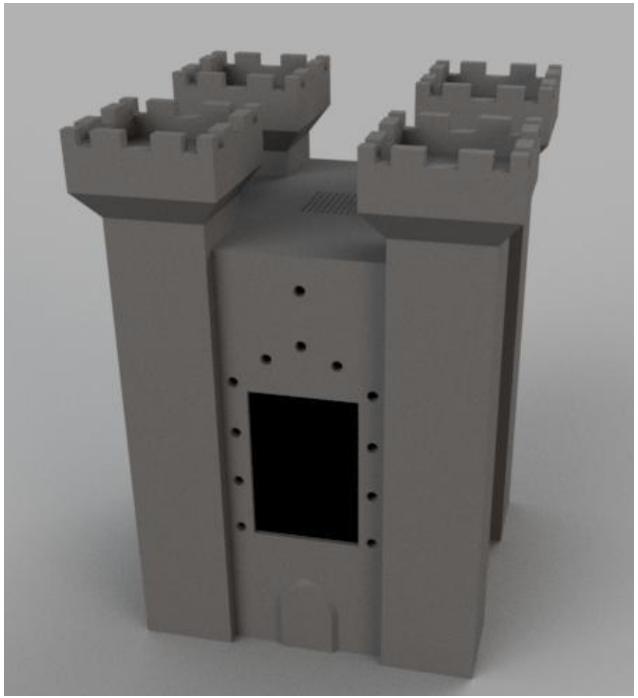


MDR Deliverables

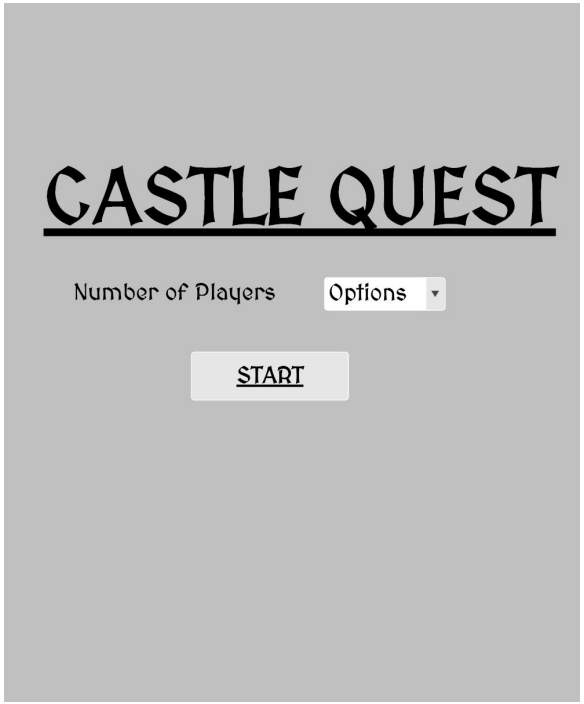
- Prototype Castle Model
- Prototype game board and UI
- Finalized gameplay
- Code system overview
- PCB design - breadboard mock-up

Prototype Castle Model



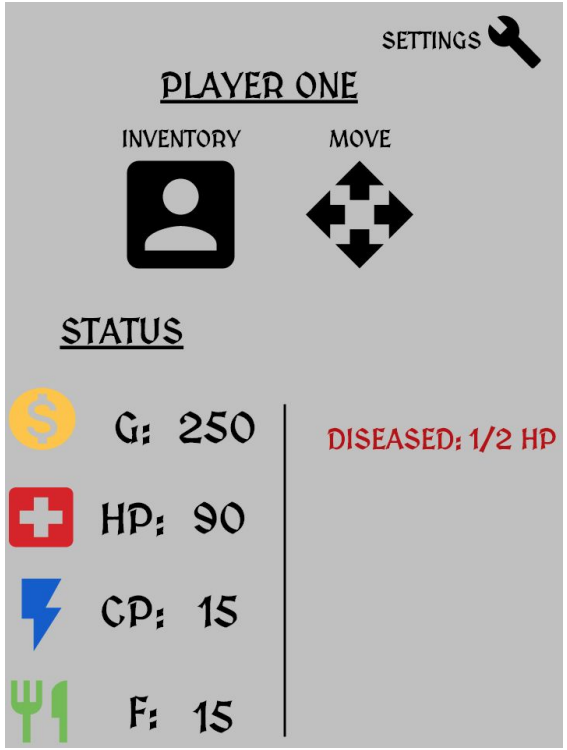


UI Design - Start Page

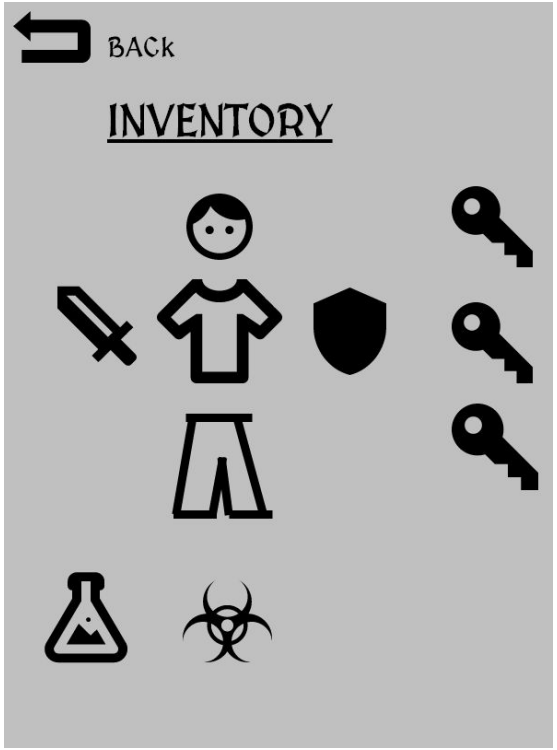


- Simplistic Interface
- Programmed using Java Swing
- Eclipse WindowBuilder plug-in, JVider plug-in

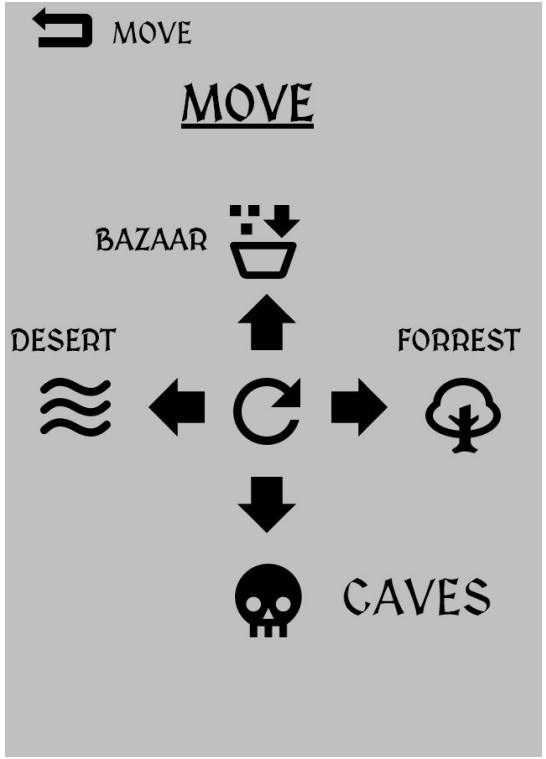
Player Start



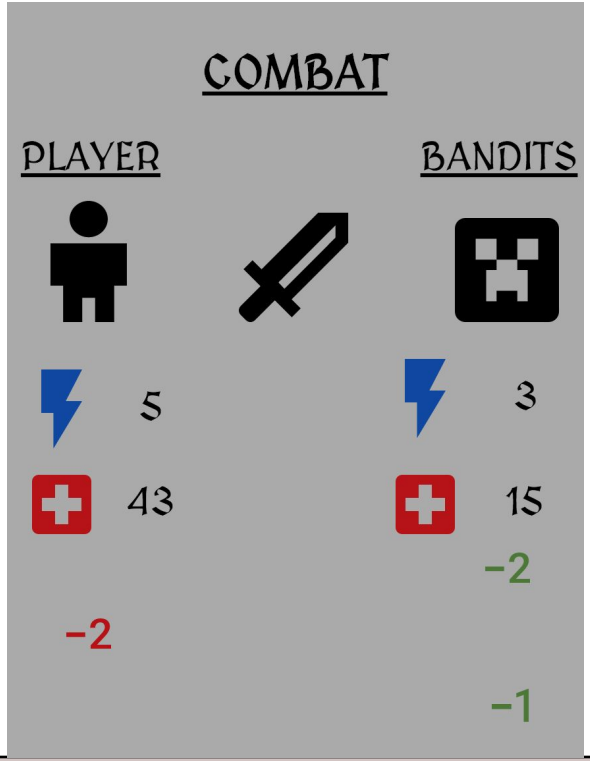
Inventory



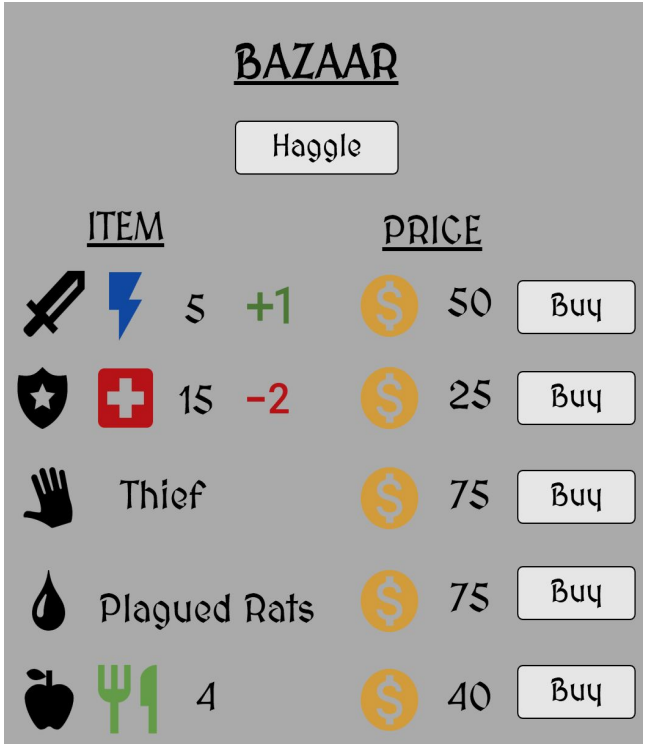
Move Selection



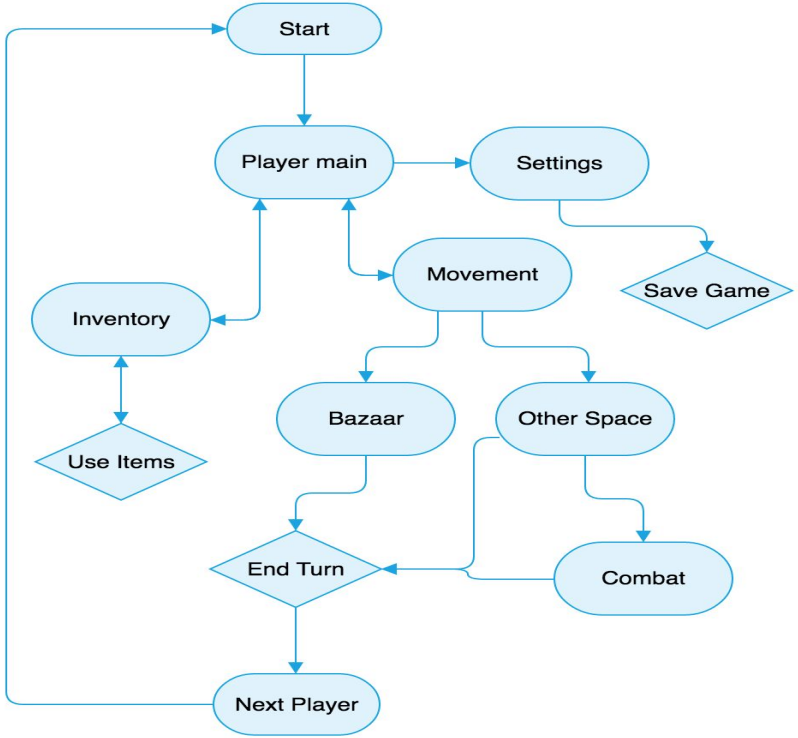
Combat



Bazaar



UI Design Cont.



- UI will be implemented by David

Finalized Gameplay

Castle Quest

Game Description:

Castle Quest is a game of fantasy adventure that takes up to 4 players on an epic journey through the four kingdoms of Derelin, Wybengaland, Lassallax, and Mangia. When traveling the lands, the players must collect three Keys of Greatness to be able to open the gates to the Castle. Through their travels the players will encounter bandits, slay monsters, fear dragons, and become heroes by defeating the evil within the Castle!

Taking a Turn:

A player's turn consists of up to three actions:

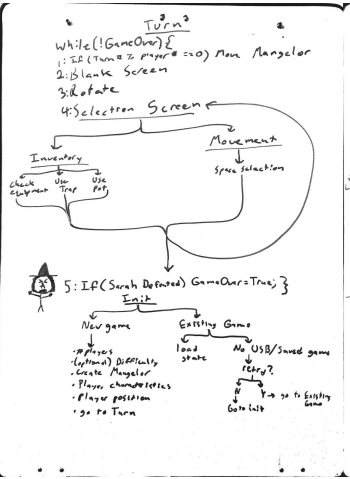
1. Checking Inventory:
   - Basic inventory consisting of Health Points (HP), Combat Power (CP), Gold (G), Food(F), and number of Keys (K) will always be displayed on the screen closest to the player.
   - Other inventory such as Weapons, Armor, Traps, and Special Items can be viewed by tapping the 'Inventory' button at the beginning of the turn.
2. Moving:
   - Players get to move one space per turn. Potential moves are presented by the screen on the tower, which the player can chose from in order to progress.
   - Each move drains one Food (except for Deserts and Mountains, which drain two).
3. Interacting with a Space:
   - Depending on the space, various Events can occur that the player has to deal with.

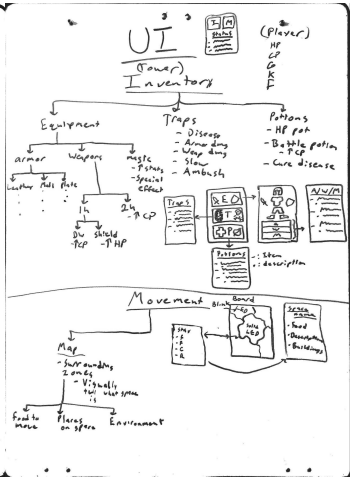
Types of Spaces:

Active spaces cause an action that the player needs to react to while Passive spaces often don't cause an interaction. However, any space besides the Bazaar and Castle Gates can be Trapped, so use caution.

Active Spaces:

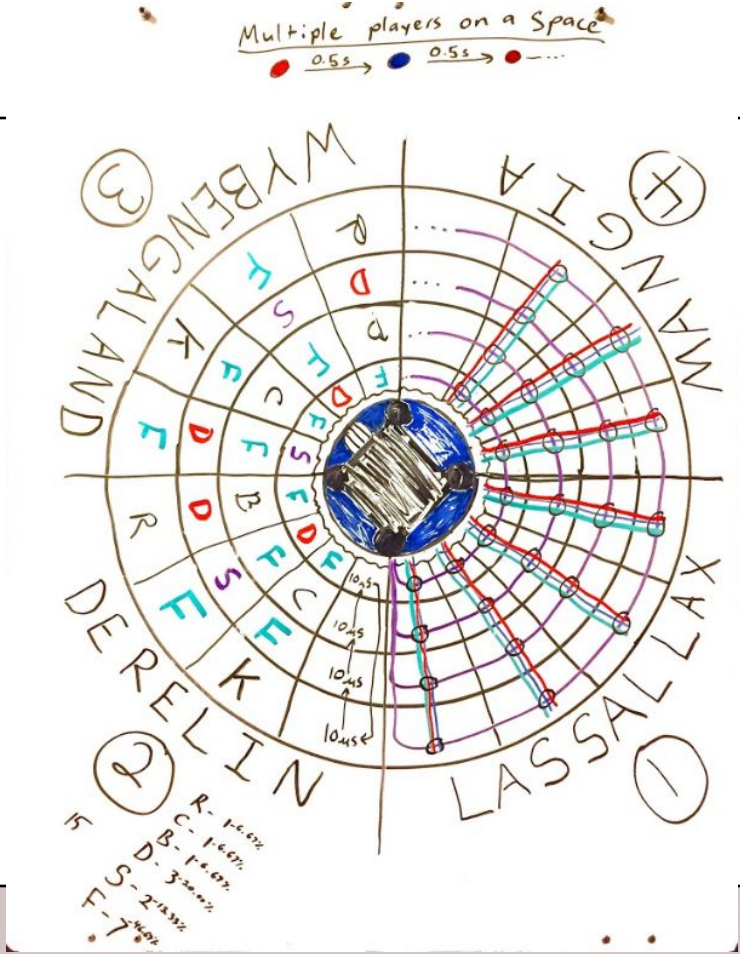
- Players Manual
- Describes rules and gameplay
- 7 Page Document





Game Board Layout

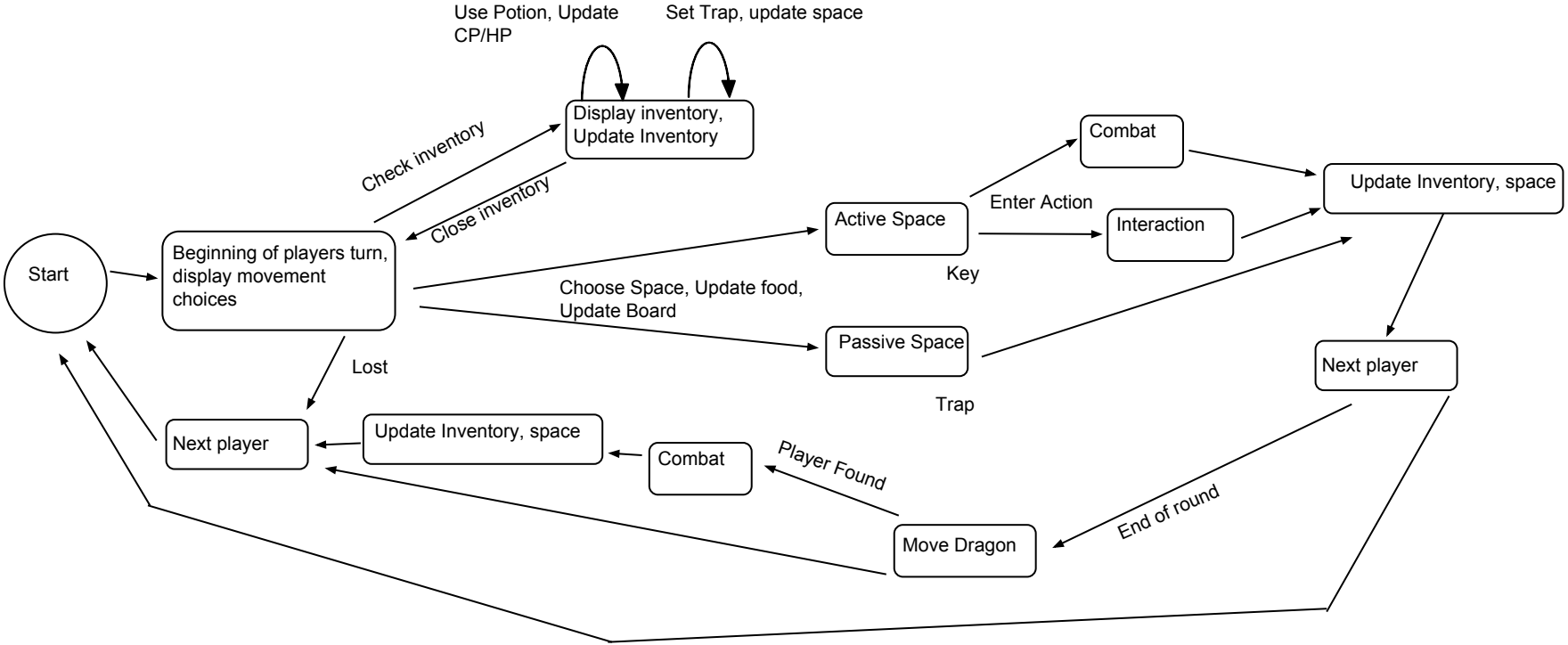
- Board will be repeating quadrants of 16 spaces
- Made from polypropylene sheet with a graphical overlay



Code System Overview

- CastleQuest java project
- Interfaces for each class with defined methods
- FSM defines game flow and will be used to build Main
- README_CODE.txt

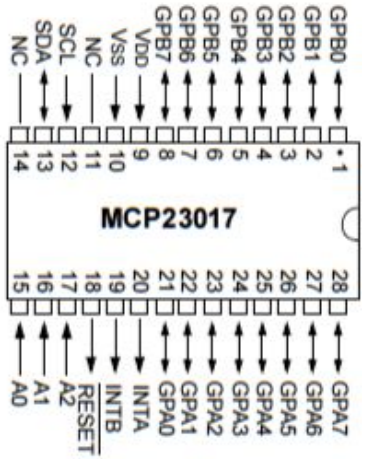
Finite State Machine

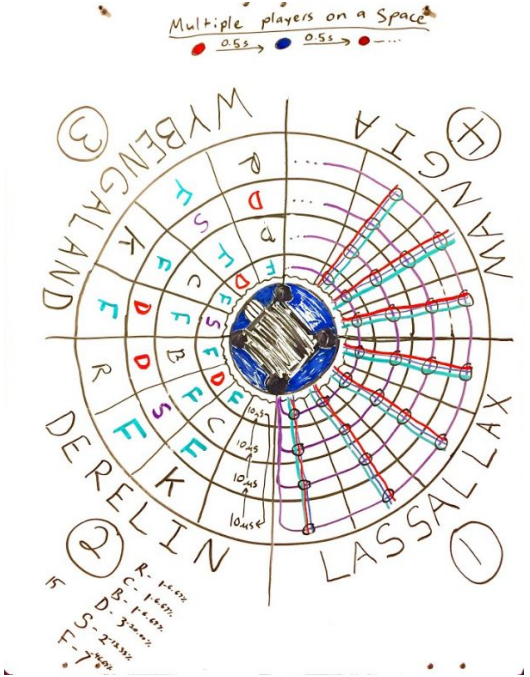


PCB Design

Handles numerous LED I/O pins

- Currently implements 6 pins for the demo
- 52 needed for actual game board
- Utilize I/O Expander: MCP23017 with I2C pins on pi



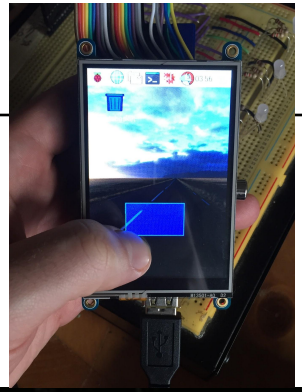


What David Did

- Sent weekly team To-Dos and organized meetings
- Designed Gameplay with Eric
- Created Gameplay FSM
- Put Castle into CAD with Fusion360
- Designed the Java interfaces with Sarah
- Designed UI

What Eric Did

- Set up Github Page
- Installed touchscreen specific OS to pi
- Configured touchscreen for use in portrait mode
- Designed Gameplay with David
- Designed board LED layout with Dev
- Debugged hardware I/O and pin usage
- Wrote LED code and prepared Demo

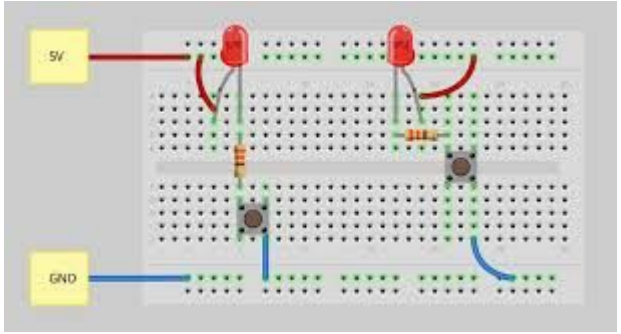


What Sarah Did

- Created java project in Eclipse
- Designed java interfaces and code outline with David
- Worked on gameplay FSM
- Ordered parts, Budget

What Dev Did

- Blueprinted Tower Model
- Designed board LED layout with Eric
- Wired Breadboard



Budget

Item	Purchased	To Be Purchased
Raspberry Pi	Free	
Castle - 3D printing		< $50
Board		$20
PCB	$1.25 x 10 = $12.50	$20
Touch Screen	$45	
Speakers		Free
Battery		$15
Memory (USB, SD etc)		$60
Total:	**$57.50**	**$195**

CDR Deliverables

- Fully Implemented Java Project - Sarah
  - Able to “play game” through command line
- PCB Prototype soldered - Dev
- Functioning LED matrix - Eric
- Printed Castle - David
- Finished UI - David
- Game Board Mock Up - Dev

Gantt Chart

	Dec 31th	Jan 15th	Jan 31st	Feb 15th	Feb 28th	CDR	Mar 31st	Apr 15th	FPR	Demo Day
Order Parts	X	X	X							
Print Tower		X	X	X						
Prototype Full PCB	X	X	X	X						
Implement UI	X	X	X	X						
Implement Gameplay	X	X	X	X	X					
Implement LED Code	X	X	X	X	X					
Construct Board			X	X	X					
Integrate code systems				X	X	X				
Print PCB				X	X	X				
Motor Implementation				X	X	X	X			
Assemble Castle							X	X		
Final Testing								X	X	X

DEMO

QUESTIONS?