MICHIGAN STATE UNIVERSITY

Project Plan
Coverage Crisis: Covering Your Assets

The Capstone Experience

Team Auto-Owners

Colin Duyck
Joe Hayes
Xander Quiton
Nik Sumnik
Andy Wilson

Department of Computer Science and Engineering
Michigan State University

Fall 2020

Functional Specifications

- Insurance is often an intimidating subject and the opportunities to learn about it are limited.

- Video games are accessible and engaging tools for learning.

- Coverage Crisis: Covering Your Assets is an insurance-based RPG where the player uses insurance to protect them and their assets from randomly occurring events.

- Coverage Crisis uses point and click style controls and is web-based.

Design Specifications

- Single Player, role-playing game
  - Schooling and Career
  - Asset Management
  - Insurance Coverage
  - Dynamic Events
- Leaderboard – summary information
- Admin Tool – in depth information

Screen Mockup: Pay Day

Screen Mockup: Player Activities
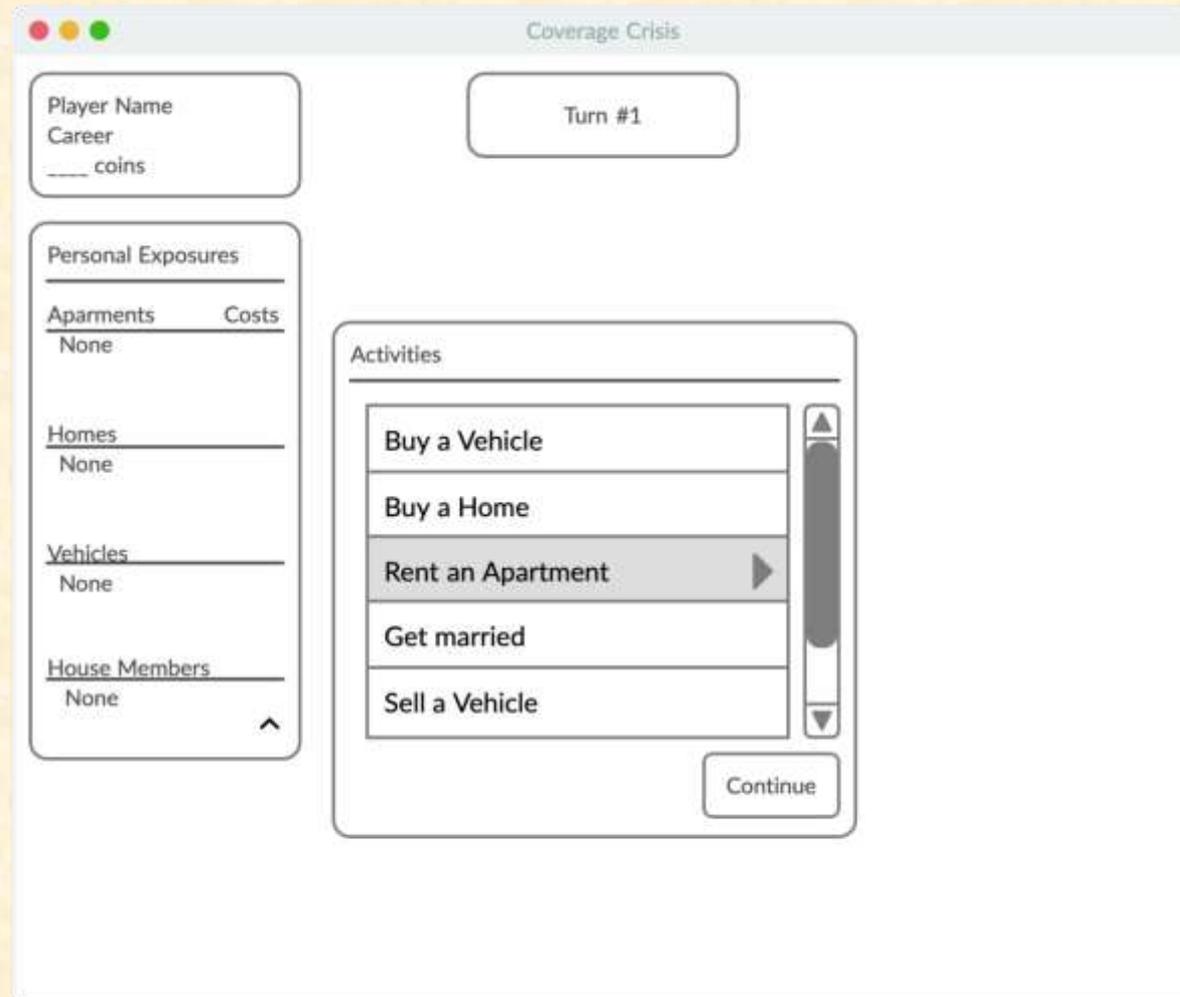
Screen Mockup: Shopping
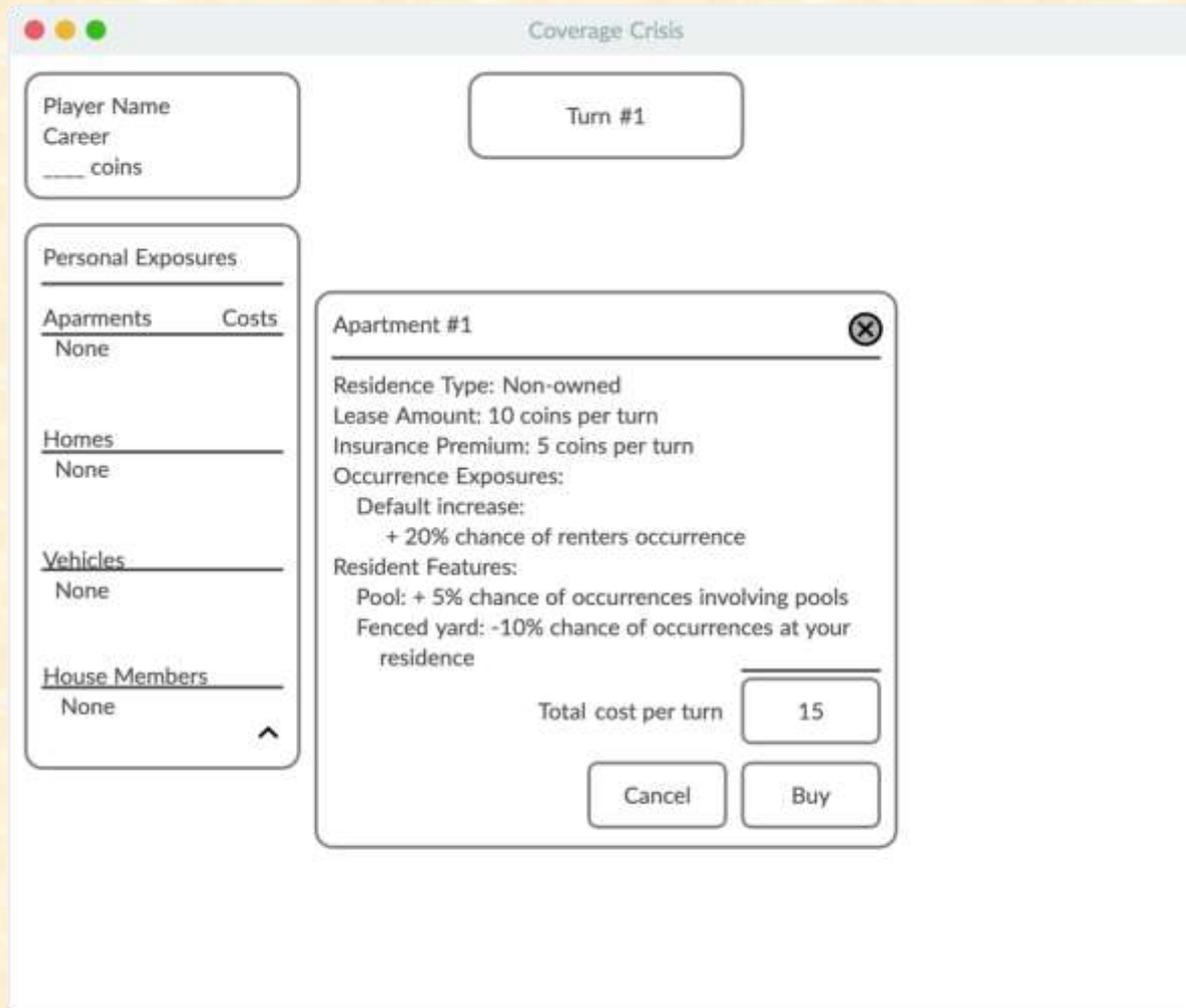
Screen Mockup: Insurance

Screen Mockup: Occurrence

Screen Mockup: Admin Tool

| Refresh				Player Search
Player Name	Date ▲	Difficulty ▲	Score ▲	Detailed Info
PL1	9/27/1776	Normal	50	▭
PL2	9/22/1776	Normal	1025	▭
PL3	9/22/1776	Hard	745	▭
PL3	9/25/1776	Easy	500	▭
PL4	9/25/1776	Normal	400	▭
PL4	9/26/1776	Normal	575	▭
PL4	9/26/1776	Hard	1000	▭
PL5	9/25/1776	Normal	890	▭

Technical Specifications

- Developed through Unity collaborating with Unity Teams

- All scripting is C#

- Database is MySQL communicated through PHP server-side scripts

- DB Admin communicates to the DB through a PHP webpage, which will be secured

- User views DB data through game application, Unity gets data from DB through the PHP scripts

System Architecture
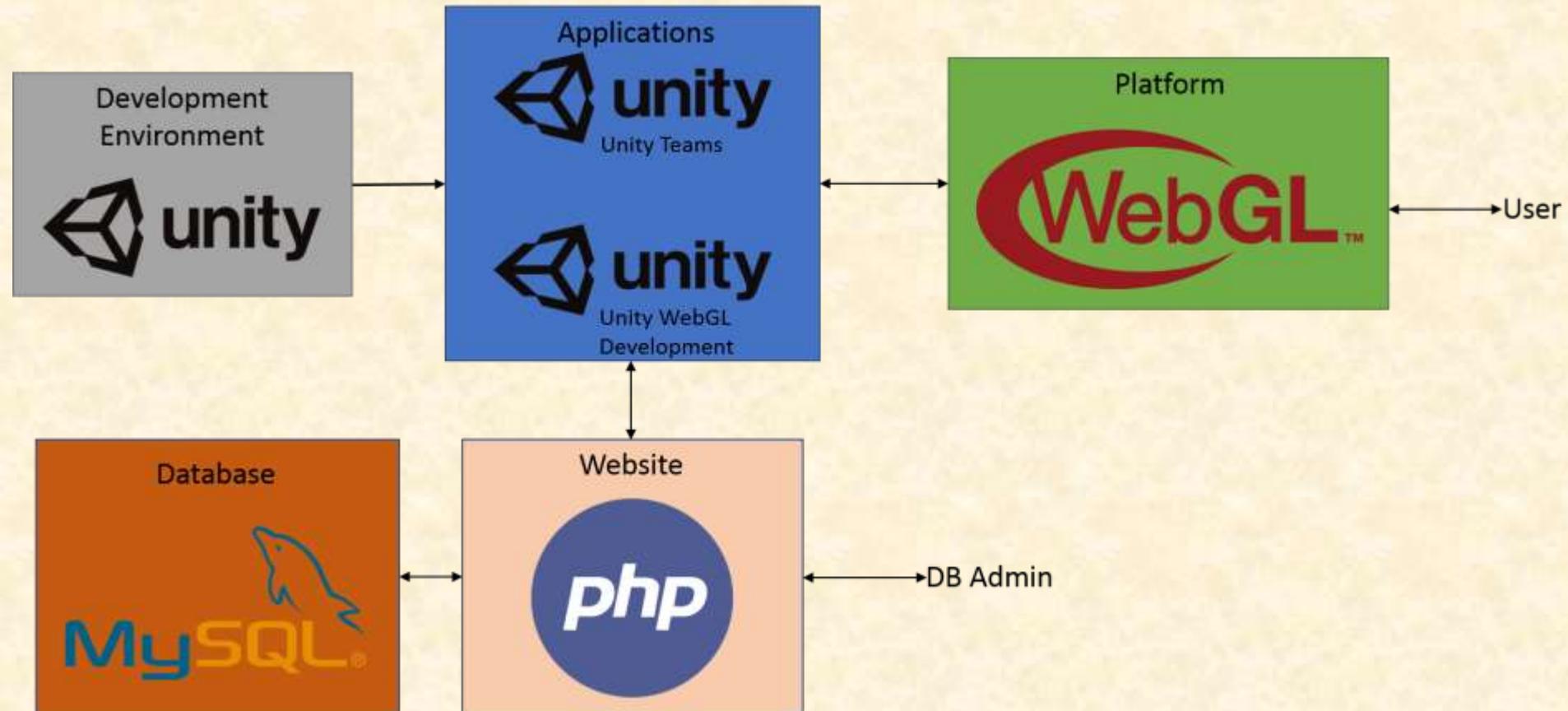
System Components

- Hardware Platforms
  - General WebGL application can be accessed through compatible browsers on compatible systems (i.e. Google Chrome compatible devices)

- Software Platforms / Technologies
  - Unity developmental environment
  - PHP - PDOs
  - MySQL

Risks

- Game Balance
  - Ensuring the game is not exploitable and can be enjoyed on multiple difficulty levels, preserving the game's identity and teaching players correct information.
  - Spending an ample amount of time to test playthroughs, and tweak data until satisfied.
- Component Prioritization
  - In order to reach the testing phase of development, we are going to need to prioritize key elements of our project with regards to scheduling.
  - By prioritizing completion of the game loop, we will enable ourselves the opportunity to get a feel for the games playability and alter specifics later.
- User Authentication
  - We need to protect our user's data, ensuring that both their connection, and our database are secure.
  - By testing methods of bypassing our login system, we will be more certain of our authentication system.

Questions?