Maverick’s Run:
An Arcade Game for Web and Local Play

[Updated Google Doc Link]

Charles V Fisher
cfisher@unomaha.edu
ITIN 8000 FA2021

Overview

Briefly describe the project and how it relates to the design brief in 1-2 paragraphs. Include sources for any figures and citations for any specific information you reference.

Students, staff, and faculty often find difficulty navigating the campus of the University of Nebraska at Omaha. Be it due to the split nature of campus (Dodge vs Aksarben), parking, or the architectural layouts, basic navigation for folks new to UNO poses a number of challenges. Historically, UNO has attempted to address this with pre-college and freshman group tours, nominally accessible maps, or wholesale altering building names and college labels. Further efforts could be done to make tours, mapping, and building identification faster, more accessible, and fun. “Maverick’s Run” approaches this in the form of a fast paced arcade game designed for play both online and on the Criss Library Creative Production Lab Arcade Cabinet.

Players will be able to navigate a two-dimensional map of campus, along with simplified structures of UNO building interiors in an effort to guide students towards quickly learning overall navigation and pathfinding across the UNO campuses.

Key Features

Create a bullet pointed list of the things the app will be able to do. This should be consistent with what you described in your overview section.

- Application will present a title screen and timer-idle attention grabber prior to interaction.
- Players can start by pressing a button on the keyboard or arcade cabinet.
- UNO campus will be rendered as a grid-based, 2-D, top down map.
- Players will spawn at an initial random location: parking lot, bus stop, or dorm.
- Players will be given a random building and room to travel to from their spawn location.
- A radial navigation arrow will point from the player avatar towards their target location.
- Players will move horizontally and vertically along the grid using WASD, arrows, or joystick.
- Players will be timed with an on-screen stopwatch from spawn to arrival at target location.
- On successful arrival at target location players can attach a 3 letter initial tag to their travel time.
- Travel times will be posted to a local-instance leaderboard.
- Players will be prompted to play again, exit to title, or timeout to title if left idle.

Minimum Viable Product Requirements

Explain how you plan to meet each of the five MVP requirements.

1. *Work without noticeable bugs.*
2. *Operate either on a cell phone or from a web browser.*
3. *Have a GUI that is navigable by someone without prior experience using your application.*
4. *Have code and documentation on Github.*
5. *Code must contain comments to the degree where a stranger could understand what the code does without running it.*

Product development will proceed in working segments. First, a basic grid map will be created and player movement and map rendering will be created using the PyGame library. From there the primary loop exit of navigating to a target location indicated on the screen will be completed. Next, a randomized player map spawn location will be developed. With those in order multiple possible target locations will be placed on the map, and one such location will be selected and indicated as the single target location. These systems in place will enable map size expansion involving a number of predefined spawn locations and target locations. Functionally, this represents the main game loop.

Additional feature expansion to make this game structure engaging relies on balancing the movement speed, placement of target locations in relation to spawn locations, and potentially defining a list of acceptable targets given an initial spawn. One challenge feature to implement will be representation of building interiors, and transitioning from an overland map of campus to a building interior. This can likely be accomplished through a single-window overlay of the building interior map triggered on building entry grid location.

Creating the radial navigation arrow sounds like the true challenge of development and needs more planning. The timer overlay mechanic can be done with existing libraries.

App build to .exe and operation on windows is of limited concern, but pushing the working application to web is something worthy of in-class or personal investigation. Early options point to this being viable.

To accomplish user navigation simplicity, on-screen arrow keys can be used as a hud overlay to indicate motion options, and the radial navigation arrow should assist the urge to make the avatar move. One option is an initial gameplay loop prompt of “Use the arrow keys or joystick to guide Maverick to their classroom!”

Personal Learning Objective Requirements

Define your Personal Learning Objectives and describe how you plan to meet them.

This application will give me hands-on experience in working outside of an existing high-level game engine in an effort to force me to use code and scripting vs pre-built (or team-built) engine tools to accomplish gameplay interactions. My primary background in working on games and interactives was in experience design and asset creation while working alongside talented professional programmers that could accomplish a shared application vision. This cooperative venture enabled our work to look and function incredibly well, but left individuals across the teams with core gaps in whole-system understanding or skill sets needed to work successfully in the absence of the full team. Namely: I could build out any feature with existing apps, or alter existing features, but needing to implement something from blank .txt to working code has been slow and problematic.

This project focuses my programming skills first developed from self-taught C# hackery and Lua, through academic training in Java, and now into a rapid-learn cycle of Python that will establish key practices and skills necessary for interactive development. Building a GUI outwards from the PyGame library vs relying on Unity's code-free tools will be a test of this new skillset.

(added 10-20-21)
An additional learning objective discussed is to gain experience in proper architecture maps. For this I plan on learning and utilizing proper UML diagramming techniques.

Personas

Who are the target users for this application (see card K of the Design Method Cards Download Design Method Cards for help)

Potential Studento, highschool senior considering UNO

- In-state student from Fremont, NE. Has been to Omaha and would like to visit campus.

Newbie McStudent, UNO Freshman

- Lives on campus. First time away from a rural home in the panhandle.

Transferino Studentstein, UNO Junior

- Lives off campus. Transferred to UNO from junior college in Columbus.

Office Stafferson, New staff employee

- Long time Omaha native, but never been to UNO's campus. Just started work at Eppley.

Teachy Facultyface, new adjunct professor for CFAM

- Just moved to Omaha, and has to teach at both Mammel and CPAC's days and evenings.

NonProfit NoMonee, new office intern at CEC

- New intern for biotech nonprofit group that just relocated into CEC and has to work alongside PKI academic stakeholders and Durham science faculty and students

(added 10-20-21)
Oma Visiterio, Omaha resident living in Midtown Crossing

- Oma has lived in Omaha for a few years but would like to come to campus for a show in WFAB, maybe a speaker at the Library, or a free-food cultural event in Milo Bail.

User Scenarios

How will the target users use the application (see card H of the Design Method Cards Download Design Method Cardsfor help).

New staff employees are tasked with working through an online orientation module (4 hours!) via Bridge that includes [currently] a video tour of UNO. At the end of this online module they are encouraged to play Maverick's Run and post their fastest time.

As part of new student orientation, first year and transfer students must complete an online orientation module. As a follow up to successful completion, MavLink sends out a notification during Welcome Week to try out Maverick's Run online to test their knowledge of getting to buildings ahead of classes.

New faculty get a tour of the Criss Library as part of their onboarding, typically from a colleague. After getting coffee at the Cafe, they stop by the CPL and play Maverick's Run on the arcade cabinet. They then stop into the CPL, ask the supervisor what the place is all about and line up tours for all their classes!

Timeline

Lay out a schedule with milestones for the development of the application from Week 8 at the latest to demonstrating your app in Week 16.

(updated 10-20-21)

Week 8

- Finalize and update PRD

Week 9

- Install pyGame and build map asset grid.

Week 10

- Map layout visible and navigable via editor. Player input working for character model in independent testspace in editor.

Week 11

- Character model integrated and input allows map navigation across the entire gamespace.

Week 12

- Randomized player start location and target location completed. Player can complete main game loop of spawn, travel, exit.

Week 13

- Radial compass working on HUD. Timer correctly tracks start-to-end of play completion.

Week 14

- Local play of executable and local storage of player record times in flat file.

Week 15

- Web playable version available for remote users to test loop and stability.

Week 16

- In-class presentation and delivery of web-accessible version of application.