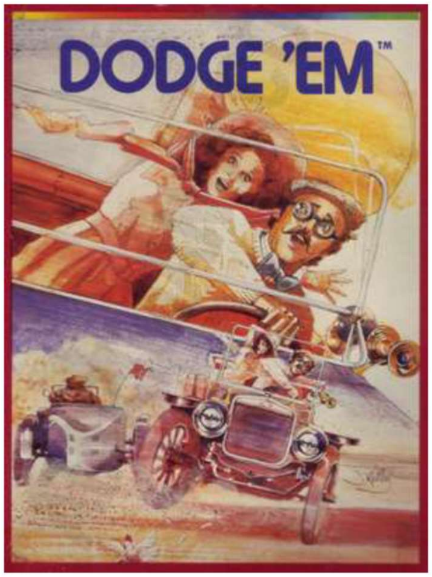
CS118: Programming Fundamentals (CSDF L&M)

Fall 2019

Project: DODGE 'EM



Open Date: 21st November 2019 at 1:00 PM
Deadline: 4th December 2019 at 9:00 AM.

Description

Dodge 'Em is a 1980 Atari 2600 driving game based on a single screen of concentric roadways. The game was programmed by *Carla Meninsky* and released by *Sears* for the *"Sears Video Arcade"* under the name Dodger Cars.

Gameplay

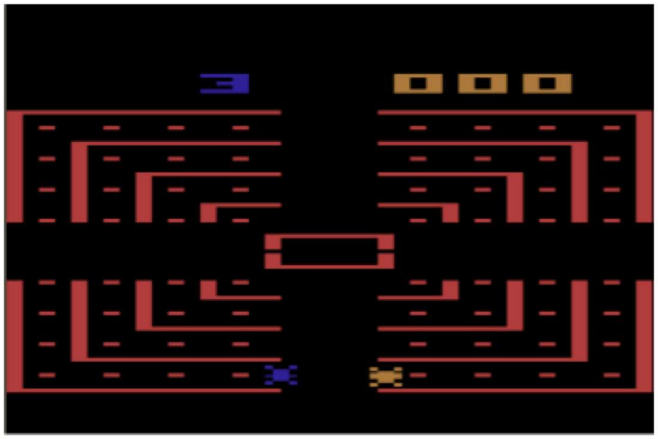


Figure 1: Game Screen

Player Vs. Opponent (45 marks)

The game is played in a racing arena where the player drives his brown colored car to pick the maximum rewards, avoiding a head-on collision with the blue opponent car(s) run by the computer (figure 1). The player car moves clockwise while the opponent car(s) move counter-clockwise only. When the game starts, the player is awarded three lives to collect the rewards from the arena. The player gets one reward point when his/her car picks up any rectangular-shaped gift-boxes placed at equal distances in the arena. When the player car hits the opponent car, one life is lost and the game arena is refilled with the reward gift-boxes. The score and the number of lives left are visible on the top of the screen (figure 2).

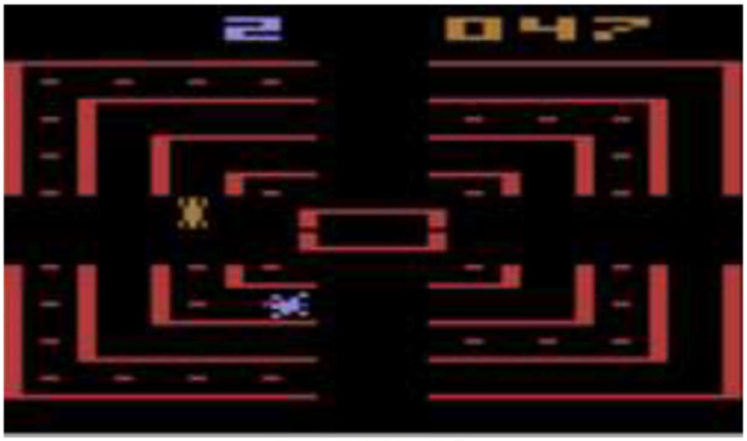


Figure 2: Driving Rules

Driving rules (15 marks)

Each roadway in the arena has four turns – one each after covering one-fourth of the roadway. The player car starts from the middle of the topmost roadway on the screen, while the opponent car(s) start randomly at any turn at least one turn away from the player car on the same roadway. The player car is controlled by the keyboard – arrow keys to turn and spacebar to drive at the double speed than normal.

The opponent car(s) makes the decision to take a turn or not on the basis of distance from the player car, i.e. it takes the turn if taking the turn brings it closer to the roadway in which the player car is moving.

Game levels (5+5+5+5 marks)

A new level starts when the player picks up all the gift boxes in the arena. The player gets 100 reward points and the arena is refilled with the gift boxes. All the cars also restart from their respective positions (as explained above in the driving rules).

Level 1: At this level, the opponent car is less active and can only take the top and the bottom turns on any roadway. However, the player can take any of the four turns to avoid the collision.

Level 2: At this level, the opponent car is in the active mode and can take any turn just like the player car.

Level 3: At this level, the opponent car is in active mode and its speed is also doubled.

Level 4: At this level, two opponent cars start chasing the player car in the active mode as in level 2. For this level, the opponent cars should not start at the same location. At the completion of the level 4, the game finishes after displaying a message, “You won!”.

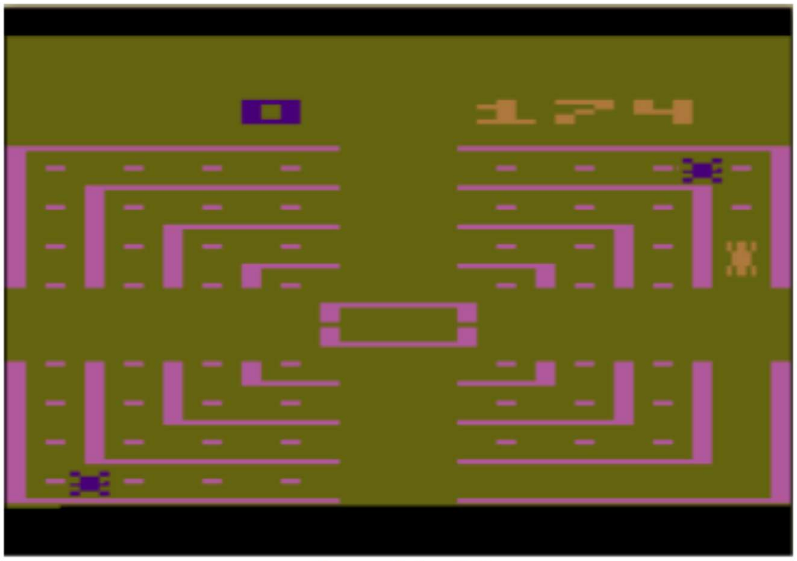


Figure 3: Level 4

High scores (10 marks)

When the game starts, the high-scores are read from a file (named highscores.txt) on the hard-disk and stored in an array of size 10. If the score of the current game makes to the list of high-scores, the lowest score is removed from the array and the new contents of the array overwrite the file.

Menu (10 marks)

The game should be menu-driven and the following options should be available to the user:

1) Start a new game
2) See high-scores
3) Help
4) Exit

During the execution of the game, the player can press the ‘P’ key to pause the game and see the following menu:

1) Start a new game
2) See high-scores
3) Help
4) Exit
5) Continue

The names of the menu items are self-explanatory and should perform the relevant function upon pressing the respective number key. In the Help menu, you can show the rules/keys to drive the player car.

Bonus (10 marks)

You may think of any interesting (and programmatically challenging) feature to implement to get bonus marks. The decision to give bonus marks will be with the teachers me to know if it qualifies as a bonus or not.

Starter Code

I will provide the basic skeleton code of the game in C++ and you are allowed to change it and add functionality according to the project statement.

Submission Instructions

Combine all your work (solution folder) in one .zip file. Use the proper naming convention for your submission file. Name the .zip file as **ROLL-NUM_SECTION.zip (e.g. 19i-0001_L.zip)**. Submit .zip file on Google Classroom within the deadline. Failure to submit according to the above format would result in the deduction of 10% marks. Submissions on the email will not be accepted.

Deadline: The deadline to submit the assignment is 4th December 2019 at 9:00 AM. There is no late submission for the project as we will conduct demonstrations right after the submission.

Plagiarism: Plagiarism cases will be dealt with strictly. If found plagiarized, both the involved parties will be awarded zero marks in the project, which means an F grade in the course. Copying from the internet is the easiest way to get caught!

Comments: Comment your code properly. Bonus marks (maximum 10%) will be awarded to well commented code.

Tip: For timely completion of the project, start as early as possible.

Note: Follow the given instructions to the letter, failing to do so will result in a zero.

References

Visit the following links to play and learn more about the game:

https://en.wikipedia.org/wiki/Dodge_'Em

https://atariage.com/manual_thumbs.php?SystemID=2600&SoftwareLabelID=144&ItemTypeID=

http://javatari.org/?PAGE_BACK_CSS=rgb(188,179,143)&ROM=http://www.atarimania.com/2600/dumps/dodge_em.zip