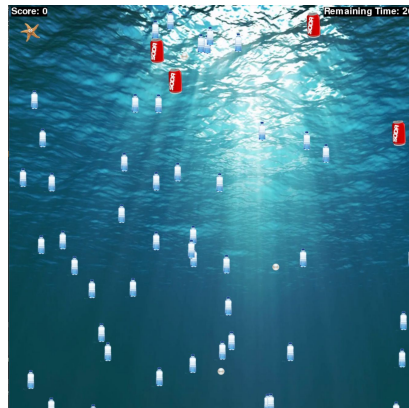


Final Catapult Report - Team 10 - Rising Tides
Pearl Catch



Owen Greybill, Brother Rice High School
Annelise Kezdy, Sacred Heart Academy
Niharika Misal, Redwood High School

18 July 2019



Project Summary

Pearl Catch, Team 10

Group Members:

Niharika Misal, Redwood High School, San Francisco
Annelise Kezdy, Sacred Heart Academy, Louisville
Owen Greybill, Brother Rice High School, Chicago

Our team, Team 10, created a computer game called Pearl Catch using the coding language Python. After learning some basic coding, our group was ready to begin brainstorming ideas for our project. Initially, we intended to create a spin-off of the classic game Snake, in which a player guides a snake depicted by a single box to “eat” food, which adds segments onto the end of the snake. Our team wanted to use an ocean theme and change the game by adding new obstacles to the game, such as plastic water bottles, which a player would have to avoid. This theme also connects to our group’s goal of spreading awareness about the deplorable state of the world’s oceans. As we began the process of coding for our game, we realized the code necessary to produce a similar game to Snake would be too difficult to replicate given our time constraints. After consideration, we decided to create an action game with a similar ocean theme.

Pearl Catch was the new game we decided upon. We started the process by naming our classes, a template definition of the methods and variables in a particular kind of object, and defining their functions. We decided to change the main character from an eel to a starfish because eels have a definitive direction in which they travel while a starfish image can travel easily in any direction without changing the direction orientation of the image. The starfish became our first class. Next, we added the class for water bottles, which would be spread across the screen as obstacles for the starfish to avoid. We then added the pearl class, which would allow the game to have an objective. The player’s goal is to move the starfish around the water bottles to collect pearls to add points to the score. Upon completing this part of the code, we

realized that our game was stationary and lacking complexity. We decided to add flare by introducing moving soda cans with random speeds and movements to give the game an unpredictable nature. A problem arose when we played our game and watched our soda cans move forward and off the screen, never to return. Upon recognizing this problem, we faced the challenge of solving this while maintaining the linear path of the soda cans. We decided to imitate a feature in Pac Man by allowing the soda cans to go off the screen and reappear with the same linear motion on the opposite side of the screen. This solution was then applied to the starfish itself to allow for more interesting maneuvers.

Upon completing this portion, we created the base of the game. We began to enhance the game by creating sounds for impact, adding a timer to allow the player to see how much time he or she has left, and displaying a “You win” screen if the player survives the 30 second game and a “You lose” screen if the player comes into contact with any obstacles. On Tuesday afternoon of the final week, the Computer Science project groups tested each other’s games and gave tips for improvement. One suggested idea was to add power pearls, a pearl that would appear rarely, but allow a special occurrence for a short amount of time. We decided to give power pearls the ability to freeze all soda cans for five seconds upon being collected by the player.

Throughout this process, we learned how to effectively communicate and listen as members of a team. We also realized the importance of compromising. The process of creating our project was made easier as we stated our goals for the day before we began coding. This project also taught us the basics of computer programming with Python, which some of us had never encountered before. The project encouraged us to grow as members of a team in order to create the most advanced project possible with what we learned.

Final Report

Introduction

Before we even began to choose our projects, we were urged by counselors and our families to get the most out of Project Catapult. We chose the computer science project to learn about programming and learn what coding would be like in the real world. Dr. Fisher worked with us to set up and learn the basics of Python through the pygame extension, which allows for the creation of games. As a class, we worked our way through harder tasks and began to feel comfortable with the syntax and the coding process. After three days of instruction, we began choosing our groups and our projects. Team 10 formed our group on the basis of a vague game idea, but we began to alter our game intentions during the brainstorming process. After brainstorming and considering our level of knowledge, we began programming the game that would become Pearl Catch.

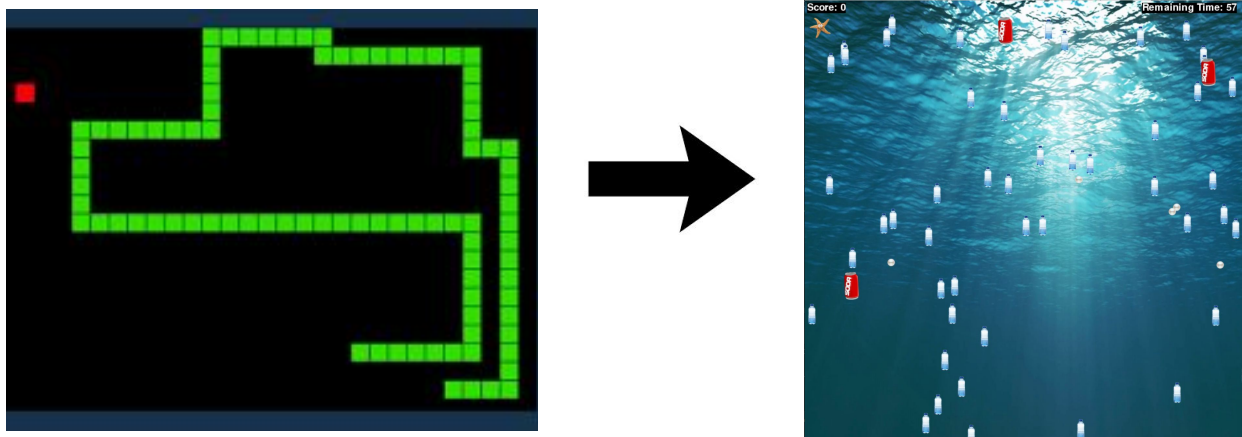
Process

Our first step in the creation process was brainstorming. On day one, we considered different styles of games, such as action or strategy. Action games involve more movement of sprites and are generally faster-paced than strategy games. Conversely, strategy games require the player to think more before moving and solve a puzzle in order to win. Our first idea was to create a strategy game in which the player solved a mystery. However, we decided as a group that we preferred an action game because they are more interesting and challenging to code, and we believed we could create a very unique action game. We searched the Internet for interesting topics and ideas, and eventually we decided to try to code a spin-off of the classic game Snake. In Snake, the player controls a small rectangle, which is the head of the snake, and moves it to eat food. As the player catches food, the snake grows in size by adding more rectangles behind the head. If the snake hits a wall or itself, the game is over. The goal of this game is simply to eat and increase the snake's size as much as possible before dying. We wanted to do an ocean-themed version of this game with a meaning or purpose behind it. Instead of a snake, we wanted to have an eel, controlled by the player, swimming through the ocean to catch food and increase the eel in size based on how much food the player catches. Rather than one level as in the classic game, we wanted to add multiple levels to make the game more interesting. We also planned to add moving obstacles, depicted as trash in the ocean, to make the game more challenging.

After selecting this game and planning some initial steps, we began coding. Our first step was to import the necessary libraries into our game. Libraries contain code that has already been written and tested and allows the user of the library to call certain functions without creating new code for them. An example would be the random library, which, after imported, allowed us to code an integer to be a random number within a set limit. Once we loaded the libraries, we considered our most important issues and decided that our first task, and the hardest task, would be coding the eel to move and adding the new boxes onto the end when the eel caught food. We created classes for different major aspects of our game. Classes provide a way to organize code when creating objects by allowing the programmer to separate the object's methods and member

variables. Methods are what the object can do, and member variables are key pieces of information the object needs to complete those methods correctly.

Our group created basic setup for the game, and as we began the task of coding the eel we realized we did not know how to move the eel or how to add to the end of the eel. Snake, the game we were attempting to replicate in a new way, is based on a grid system, in which the snake is actually a number of boxes on the grid lit up in a different color in a particular pattern based on how the user moves the snake's head. Our knowledge of Python is limited, so after trying to figure out this grid method, and getting help from our TAs and faculty member, we decided to reconsider our game. We discussed having a free-moving eel, unlike Snake, so we could use our knowledge to create the game in a timely fashion. Although we wanted to face the challenge of learning new code to create our game, we decided as a group that we did not have the time necessary to code our original idea.

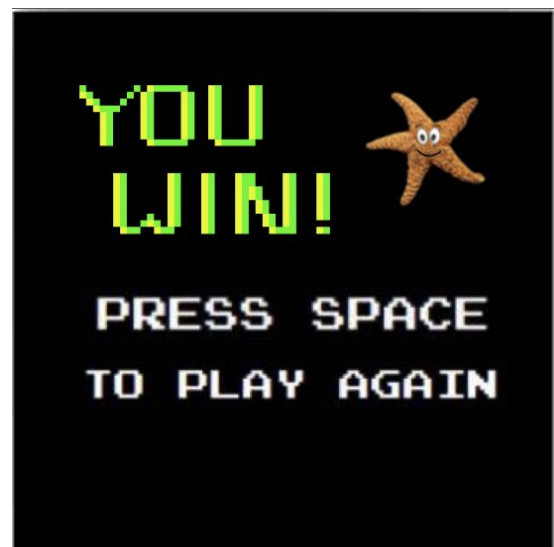


Our ideas changed from a game similar to the classic Snake game (left) to our final, free-moving Pearl Catch game (right).

As we worked on a free-moving version of the game, we decided that the main sprite the player moves should be a starfish because it can move in any direction without having to re-orient the picture. Rather than having the player guide the starfish to food, we decided to add in pearls as the goal for the player to collect. We created classes for each of these groups and decided to have three pearls spawn randomly on the playing screen. To create the challenge of the game, we set images of water bottles to spawn randomly on the screen, which the starfish cannot touch or the player loses the game. At this point in the creation process, we routinely defined our next steps and the things we needed to code next in a “to do” list. We selected items off the list based on what made the most sense to create a working game and tried to add problems to the list as they arose.

Over the course of several days, we defined all our classes, coded the starfish to move freely around the screen based on the user's input, set the pearls to spawn, set the water bottles to spawn randomly, and decided to add more obstacles to make the game more interesting. We decided we needed another moving piece on the screen besides just the starfish. Working with our ocean pollution theme, we decided to have soda cans float freely around the screen at different speeds. We also coded the pearls to add 10 points to the user's score and displayed a

scoreboard in the top left of the screen. On the top right of the screen, we added a timer so there was a more urgent element to the game and so there was a natural way to win the game. We decided the goal of our game would be to collect as many pearls as possible in a set amount of seconds. If the player manages to dodge all of the trash and survive 60 seconds, they win and they get to see their score on the “You Win!” page that displays. If the player runs into a water bottle or is hit by a moving soda can, the game is over and they are given the option to play again by pressing the spacebar.



One of these images will display after the game is ended. If the user wins the game by surviving 60 seconds, they will see the “You Win!” screen on the left. If the player collides with a water bottle or is struck by a soda can before 60 seconds has elapsed, they will see the “Game Over” screen on the right. On both screens, the user is presented with the option to replay the game by pressing the spacebar.

During the project work time, there was a round of peer reviewing which allowed each group to get feedback based on user experience. Based on the comments of test users, we decided to add “power pearls,” which would have certain effects on the game to make it more interesting. We added a red pearl, which we set to spawn randomly 20% of the time a pearl is spawned. The red pearl, if collected by the user, will stop the movement of the soda cans for 5 seconds. After adding the power pearl, we tested our game again and decided we needed to prevent pearls from spawning on top of water bottles. We had already created code to prevent the center of a pearl from touching a water bottle’s hit box. However, this logic allowed pearls to overlap with water bottles in such a manner that they were uncollectable. With the help of Dr. Fisher, we fixed the overlap problem between pearls and water bottles, and then we decided to use the same code to make the collisions between the starfish and the water bottles and soda cans more exact. With our new code, the game is more difficult because the edges of the starfish rather than the center determine if the starfish has collided with the edge of a soda can or water

bottle. Our final task was adding the introduction screen so the player can load the game and then decide when they are ready to start.

```
100     def touching_waterbottle(self):
101         if self.x is None:
102             return True
103         for waterbottle in self.waterbottles:
104             if self.x > waterbottle.x - 16 and self.x < waterbottle.x + 16 and \
105                 self.y < waterbottle.y + 43 and self.y > waterbottle.y - 16:
106                 return True
107         return False
```

The image above is a screen capture of part of the code we used to resolve the issue of the pearls overlapping with the water bottles. This portion of code is a method within the Pearl class. This method iterates through the waterbottles list and identifies the coordinates of each edge of the water bottle hitbox based on where the water bottle is located on the screen. This is then used to ensure the pearls have not spawned within the water bottles' hitboxes.

Project Theme

In light of the increasing use of plastic in our world and the resulting persistent ocean pollution issue, we realized that we wanted to focus and hone in on this global issue and raise awareness through our game. With these goals set out, we began brainstorming ways to incorporate this into our project. We chose plastic water bottles and soda cans, two of the most common forms of ocean pollution, to reveal the severity of plastic pollution in our oceans. For our character, we chose a starfish as they are a lesser-known yet still very threatened marine organism. “Starfish are linchpins in the ecology of habitats like tide pools,” said Robert Paine, a retired marine ecologist at the University of Washington in Seattle, in an interview earlier this year. “Without them around to control mussels, the bivalves can take over an area, greatly reducing the kinds of algae and sea anemones present.” The breakdown of plastic water bottles results in the development of microplastics, tiny bits of plastic that are generally less than 5 millimeters in length. The extent of microplastic pollution is becoming increasingly clear and we hope that as players engage with our game, they come to realize the large impact that human plastic pollution has on marine organisms. Ingestion of microplastics by species at the base of the food web is a cause for concern as little has been known about its effects until now and the feeding behavior of starfish will likely be affected. With this important purpose in mind, we had more motivation to overcome challenges and make the game as we envisioned it,

Challenges Faced

We faced many challenges throughout the game-making process. At first, we realized that computer science is more about finding out why code does not work rather than writing code, but as the weeks went on, we learned more and more. Brainstorming was the first challenge we faced. We had a rough idea to create a Sherlock Holmes game, but realized it

would be too complicated. We then decided to replicate the Snake game but create an eel in front of an ocean backdrop rather than a snake. After realizing the complexity of this task and the limited time we would have, we decided to do a free moving game, with similar fundamentals as what we had coded as a class. The next challenge was getting the starfish to collect the pearl and respawn another. This was solved by using the function `collide_point` and creating a list of pearls. We faced trouble with the starfish spawning on top of the water bottles at the start of the game, causing the game over screen to appear instantly. This challenge was solved by limiting the region in which the water bottles can spawn to exclude the area in which the starfish spawns. Finally, we faced the challenge of stopping pearls from spawning on top of the water bottles. This was solved by checking if any pearls spawned on top of the water bottles and rearranging the screen until there was no overlap. We worked together to create solutions to any challenge that occurred.

```
number_of_waterbottles_in_region_1 = random.randint(5, 10)

for x in range(number_of_waterbottles_in_region_1):
    waterbottle = WaterBottle(screen, random.randint(0, 80), random.randint(70, 900))
    waterbottles.append(waterbottle)

number_of_waterbottles_in_region_2 = 50 - number_of_waterbottles_in_region_1

for x in range(number_of_waterbottles_in_region_2):
    waterbottle = WaterBottle(screen, random.randint(90, 900), random.randint(0, 900))
    waterbottles.append(waterbottle)
```

The image above is a screen capture of a section of code we used to resolve the issue of water bottles spawning on top of the starfish at the very beginning of the game. This spawn overlap caused the starfish to be killed instantly and ended the game immediately. The section of code above creates two regions out of the screen in which the water bottles can spawn, leaving a space in the top left corner for the starfish to spawn. The water bottles spawn randomly within their respective regions. The number of water bottles per region is random, but the number of water bottles in the smaller region 2 is based on the number in region 1 so there are never more than 50 water bottles on the screen.

Group Project Reflection

Over the course of this project, we were all able to learn more about the engineering process and the meaning of being an engineer. We learned that keeping consistent goals and checklists allowed for an effective and time-efficient approach. The engineering mindset involves motivation and an open-minded outlook on challenges that arise. We each brought different ideas and engineering concepts which allowed for a free-flow of thoughts and solutions. With our various skills, we were able to devise several approaches to our problems and test our different ideas. We realized that engineers must be able to understand complex ideas and ask questions about concepts and processes. Often times, engineers face unexpected challenges and they must be able to identify them and fabricate several solutions to test. We were all able to acquire the basics of an engineering mindset and reflect on our growth as engineers.

In addition to having a work-ready and ambitious mindset, each member of our group brought a unique skillset to the table. Our team was very balanced because each member was flexible about work and compromising and we all had different areas in which we excelled. Niharika had extensive knowledge of picture-editing software, so she was able to work with and modify any image we wanted to use in our game. She was also good at encouraging the group to try new methods to solve problems and in proposing solutions that effectively addressed our issues. Owen was excellent at typing up the code for the group as we all brainstormed ideas and he brought several ideas to the group that solved several problems we faced. Annelise was good at brainstorming solutions to coding problems we faced, helped keep the group up-to-date on turn-in deadlines, and worked hard on the written parts of the project. She was excellent at carrying out our solutions and in making sure that we eliminate any sources of error that would arise in the future of our code. We all had to work hard to develop our Python skillset when we arrived at Project Catapult because none of us had very much experience with coding in Python, but we were able to stay focused and motivated throughout the Python classes so we could each bring some basic knowledge of the coding language to our group work.

We learned more about teamwork during this project. We had experience in our high schools with team projects and sports, but this was the first long-term project that we as individuals worked on. We realized early on that we were more creative and better at problem solving when working together than individually. Although we did break certain tasks into individual parts, most of the project consisted of us all working on the same task to solve the problems as they occurred. Typically, we all centered around one monitor while one person typed and all three of us brought up ideas and syntax to accomplish the goals we wanted with code. Shijun and Derek, both teacher assistants with incredible knowledge in the field of computer science, as well as Dr. Fisher made this project achievable. With the insight they have earned over many years of coding, they helped us find solutions when we were stuck on our own. In addition to the three of us coding, Niharika edited all of the images and text that appear in the project. Owen effectively brainstormed many of the ideas of the project and brought plenty of laughter to the group. Annelise worked hard on the poster and written reports and provided many unique solutions to difficult programming errors. The work ethic and coordination of our team allowed us to finish with the game we did. We bounced ideas off of each other constantly and all worked hard to accomplish our goal of providing a game with a message that adds joy to the player's life.

Individual Project Reflections

Owen Greybill

I came into Operation Catapult with no coding experience but with an interest in computer science. After completing a game in Python, I would never have thought I would learn so much in a short period of time. I realized that programming has more to do with creativity and problem solving than I originally realized. My favorite part of this project was the interactions with my team members, Dr. Fisher, Derek, and Shijun. Their knowledge of coding shows what hard work and time can achieve. Our project developed into a completely different idea. I learned how to code and develop a game as a part of a team. With an interest sparked from

lightning lectures and conversations with my peers, Operation Catapult has proven that an education in the field of engineering and science is right for me.

Niharika Misal

Looking back on my experience at Operation Catapult as an engineering student, I feel I have successfully built a foundation for future engineering opportunities. I have expanded my knowledge of engineering and gained an understanding of coding that will help me to further develop these skills. Beyond coding skills, I have also learned that I want to pursue engineering and computer science in my college career. This experience allowed me to understand how to work in a team and to facilitate a conversation about addressing problems. I was also able to gain a better understanding about dividing tasks and how to identify potential errors that may arise in the coding experience.

Annelise Kezdy

Before attending Operation Catapult, I had very limited experience with Python. I had taken some computer science classes previously and I was familiar with various basic coding concepts, but my classes had not taught me Python to the extent I learned at Catapult. I have also grown in my ability to work as part of a team and contribute by using my skills in coordination with others. Each member of my group had certain skills and prior knowledge walking into this project, and through compromise and coordination, we worked effectively as a team to create our desired product. Owen was a very good typist and was very organized. He helped keep us on track by creating to do lists and helping us define our next steps. Niharika was great at suggesting new coding methods to solve problems that arose. Her creativity also helped us create a game that looked nice because she helped pick out good background images and pictures for sprites. Her knowledge of image-editing software also helped our group because she was able to edit our chosen images so we could use them for our project. I helped my group by staying on task and suggesting fixes for our coding problems. I had limited knowledge of Python, and I used this knowledge when suggesting ideas. I also helped with the majority of writing tasks.

My team overcame challenges by planning. If our group encountered a challenge while coding, we stopped and planned how to fix it before typing our code. We each used our skills to guide the group to find a solution. We also worked together well because we could communicate well. We listened to each other and compromised while creating our game. I believe my mindset helped during this project. I tried to keep a positive attitude about encountering challenges and view them as a chance to increase my knowledge. I also kept myself motivated by considering the different ways we could improve our game and keeping in mind how much I would learn by overcoming the next challenge or solving the next puzzle. Overall, my group worked together well and our finished product shows this.

Conclusion

In conclusion, our team worked effectively together to learn Python and create our game, Pearl Catch. We each had our own skillset we brought to the table of prior experience and areas of expertise, but we all had to learn Python together, so we helped each other with different areas of that particular skill. Every member of our group kept up a positive and ambitious attitude,

which allowed our group to make the best project we could considering our knowledge of Python and our limited time. We did not balk at taking on new challenges to add to our game or improve our code, and we worked efficiently to make the most out of our time. Overall, Team 10 created the best game we could and the experience taught us the basics of Python and important lessons in teamwork and time management. We are looking forward to future engineering opportunities in computer science and software engineering and are eager to advance our knowledge in the coming years and to see how engineering shapes and changes our world.