Authors: Aaditya Raj, Ishaan Singh, Rajiv Venkatesh
Revision: 5/23/22

Introduction:

Our program is a 2 player game with a side view. In this game, there is a battlefield, which consists of platforms to jump on. Players start from their home bases, one in the bottom-left corner and the other in the bottom-right. Players have the ability to navigate the map and jump up onto platforms to eventually reach the flag. They each have a paint gun, with which they can shoot paint onto the platforms or shoot the other player and eliminate them. If there is already paint in the spot, nothing will change. The paint on the ground gives the player the ability to move 2 times faster if they 'swim' in it by passing through. When the player shoots their opponent, they lose health, and if they are eliminated, they respawn back at their home base. At the top-center of the map there is a flag, which also serves as a stronger paint gun. The goal is to take the flag and go back to your base, which is how you win. When someone takes the flag, they can use it to shoot paint with a bigger stroke width and damage their opponent more; however, it shoots less frequently than a regular gun. Furthermore, the person holding the flag cannot move faster on their color paint. When someone dies holding the flag, the flag will remain in the death location. In addition, there is a 2 minute time limit. At the end of the time limit, the win is awarded to the player who has the most points. Points are a combination of the percentage of the map covered in the player's color of paint, the number of flag captures, and the number of deaths. The number of points awarded for paint percentage is multiplied by 1000 to produce a value of equal weight with deaths and captures. Each player can be shot 7 times before the die and respawn, which results in the loss of 200 points. Similarly, a capture of the flag results in an increase of 200 points.

People who would love our program are those who enjoy games such as Splatoon or Capture The Flag. This game merges both in 2D, so the game feel and mechanics are altered completely. Gamers who play Halo 5, Call of Duty or Fortnite are also likely to be fans of this game, as they already play games in which aiming at the right target with the gun is the most important goal.

One primary feature of this program is the ability to shoot paint at the ground to 'swim' in it for faster speed, to damage the opponent, and to gain points. Platforms are another major feature that enhance the user experience by making the game more difficult and interesting. The flag is another crucial feature. This object serves a dual purpose – the player has to capture it and bring it back to their base to win, but can also use it as a stronger paint gun. Lastly, home bases are useful for the respawning of the player.

The reason all three of us were interested in making this program is because we all love multiplayer games where we play against each other and we all like the idea of capturing the flag, and the idea of a territory battle with paint.

Final Release Instructions:

1. Right-click the jar file to run the program
2. If creating a room, follow the instructions below
    a. Enter your username
    b. Click create room, and wait for your opponent to join the game
    c. You will show up on the left side of the screen
3. If joining a room, follow the instructions below
    a. Enter your username
    b. Enter the IP address of the hosting player
    c. Click join room, and you will show up on the right side of the screen
4. Once you are in the game screen
    a. Use WAD to move jump, move left and right
    b. Click the mouse to shoot
    c. When your gun increases in size, you shoot paint bombs
5. After 2 minutes, the win screen shows who has won
6. Quit the game and restart to start a new game

<u>Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER)</u>:

Must-have Features:

[These are features that we agree you will *definitely* have by the project due date. A good final project would have all of these completed. At least 5 are required. Each feature should be fully described (at least a few full sentences for each)]

- Multiplayer function
    - Arrow keys for one player, WASD for the other
- Map with platforms, barrels on platforms, wall border
    - Restricts both players' movement
    - Cannot swim over barrels, must jump over
- Running and 'swimming' through paint
    - Regular running - walking on non-paint territory or your color
    - Swimming in your paint faster than regular walking
    - If you walk on the opponent's paint, you get 'dizzy': slow down.
- Paint-shooting
    - Try to cover the platforms, walls, barrels with your color paint
        - Determines win if no one collects the flag by the end of time limit
    - Three hits to kill opponent
    - Automatically reloads (time-based)
    - Reload the paint gun faster by 'swimming' through your color
- Flag
    - Goal is to reach the flag at the center and bring it back to your home base.
    - Flag lets you shoot bigger paint strokes, but less frequent shooting, and slower movement

        - 3 hits to kill opponent
- Home bases
    - Base where you start and where you respawn after dying

Want-to-have Features:
[These are features that you would like to have by the project due date, but you're unsure whether you'll hit all of them. A good final project would have perhaps half of these completed. At least 5 are required. Again, fully describe each.]

- Multiplayer function with networking, two different devices
- Choosing different paint guns with different reload speeds, paint stroke width, paint stroke length, etc. before the match
- Each player starts with a few paint bombs: items that they can throw and they will burst with paint covering a certain radius
    - Paint burst does not pass through walls/platforms
    - Select how much each player starts with when creating a game
- Deploying small towers that shoot paint around itself anonymously in a specific pattern, can damage the opponent, prevents the opponent from passing through and it has less health.
    - # of towers per player would be set when creating game
- Choose out of a few maps to play when creating a game
- Leaderboard
    - The top 10 users are displayed from the server and it shows the max points they received in a game.

Stretch Features:
[These are features that we agree a *fully complete version of this program would have, but that you probably will not have time to implement*. A good final project does not necessarily need to have any of these completed at all. At least 3 are required. Again, fully describe each.]

- One stretch feature we could implement is 3D graphics. This would require a number of extraneous classes, which we felt would take up valuable time that could be spent improving the user experience.
- Another of these features is the first person perspective. As of right now, we do not know where to start with this, and although it would be a very cool addition to our project, we are fine with the perspective we have now, which will also make debugging easier.
- Finally, we hope to be able to incorporate more than two players in the game, but this would require a larger map and likely more complex networking.

<u>Class List</u>:

- Main
    - Runnable class that creates drawing surface
    - Uses DrawingSurface
- DrawingSurface
    - The drawing surface on which the game is displayed

- Various Screen classes
    - Each screen that the user sees
- MainMenu (extends Screen)
    - Beginning screen of the game
    - Where the player is introduced to the came
- CreateRoom
    - Creates a room for the game for another person to join.
- JoinRoom
    - Another player joins the room by entering the IP address of the host's room in order to play the game against the host.
- GameRoom (extends Screen)
    - Where the player can customize and create a new game for another person to join
        - Set a time limit, map, etc.
- Game (extends Screen)
    - Displays the actual game
    - Tracks score
    - Displays the remaining time for the game.
    - Shows the player's total health.
- Platform
    - The platforms that each player can jump on to reach the flag
    - Has a PaintBlock border to represent the parts that have been painted
- Flag (extends PaintGun)
    - Is a paintgun, but there is only one in the center of the map
    - If one user is able to capture the flag and return it to their home base, the game ends and that player is heavily favored to win
- Avatar (extends Sprite)
    - Represents the player on the map
    - Health, speed, color
    - Jump, move, throw bomb, shoot paint
- PaintGun (extends Sprite)
    - int ammo, reloadTime, velocity
    - PaintBlock stroke
    - shoot(int x, int y)
- PaintBomb (extends Sprite)
    - A bomb that the player can throw
    - Has a certain radius of explosion, releases many PaintBlocks when exploding
- PaintBlock (extends Rectangle)
    - Is a square, represents a 'block' of paint used to paint the map
- WinOrLoseScreen
    - It displays the win screen for the winning player, lose screen for the losing player, and it would display a tie game screen if the two players tie.

Bugs (* Important to Note ***) :**

- You can only have one window open, otherwise networking will crash
- If somebody closes the window while playing, then a socket exception is thrown
- Slight score discrepancy
- You may have to click a few times for create and join button to function

Credits:
[Gives credit for project components. This includes both internal credit (your group members) and external credit (other people, websites, libraries). To do this:

- List the group members and describe how each member contributed to the completion of the final program. This could be classes written, art assets created, leadership/organizational skills exercises, or other tasks. Initially, this is *how you plan on splitting the work*.
- Give credit to all outside resources used. This includes downloaded images or sounds, external java libraries, parent/tutor/student coding help, etc.]

External Libraries

- Processing and G4P
- Networking libraries

External Assets (edited using Adobe Photoshop)

- **Background** (via dribble)
- **PaintGuns** (via 123rf)
- **Avatar** (via dribble)
- **Win/Lose Screens** (via Adobe Behance)

Rajiv

- PaintGun & Flag
- Set-up (Main, DrawingSurface, Sprite import)
- Game
- Images & other resources
- JoinRoom
- CreateRoom

Aaditya

- PaintBlock, PaintBomb, Platform classes
- PaintGun
- Avatar
- Game
- Server networking
- JoinRoom
- CreateRoom
- MainMenu

Ishaan

- Avatar class
- Game
- WinOrLoseScreen class
- Platform
- Images & other resources