Project Summary: Medieval Bashout, Team 13

Group Members:	Richard Caccamise	Tanque Verde High, Tucson, AZ
	Nicholas Shircel	Plymouth High, Plymouth, WI
	Alexander Trick	American School of Doha, Qatar

Over the past week, our team has developed a game titled Medieval Bashout using the popular programming language Python. To start, we had a few days of instruction in order to understand the basic principles of the language, as well as some time working with the *pygame* library and PyCharm. Then, teams were formed, and began brainstorming ideas for projects. Super Smash Bros. was an immediate source of creativity for our group and we took inspiration from many of its features in our game. Multiplayer functionality is a key part to Super Smash Bros. so our original goal was to have 4 players playing simultaneously across four different computers. However, we quickly realized that we would have to simplify our game significantly in order to finish on time.

After deciding that we would start with a 2-player game using a single keyboard, we split up to tackle different elements of the project. Members of the team began work on the art, including the background images and character sprites, while other members began the coding work. We started by defining a few classes such as *DamageCounter*, which would later control and update each player's health. We then created the *Player1* class, the wizard, and started to define methods such as *draw()* and *hit_by()*. Once a frame for the wizard character was complete, we compiled code to load it into the game and display a character for the first time. We then added functions to allow the character to move left and right. Once all the frames for the wizard character were complete, we immediately ran into a roadblock. There was no way to animate the character with the current method of movement. This was the first time we really had to combine the knowledge of all the members in our group to solve a problem, but certainly

not the last. Once we created additional methods in the *Player1* class for movement and set the frames to be displayed every five frames in the game, we had a working, animated character. We reused a lot of this code in the creation of the *Player2* class, or the knight, as they are fundamentally the same character.

Now that we had two working characters, a background, and damage counters, it was time to have them start interacting. The first task was to create some simple gravity in the game which was quickly completed. However, the task of stopping the characters when they hit platforms was much more complicated. We had to create a *Hitbox* class that would tell our characters to stop falling when they hit one of the hitboxes, which we put around each platform in the game. We continually ran into issues until we completely revamped the class by replacing the *pygame.collidepoint()* with the *pygame.colliderect()* function so that the entire character would be considered instead of a single point. We ran into additional issues with jumping, as you could hold down the key and fly to the top of the screen; animating jumps and slash attacks, as they would only change frames every five times you hit the key; calculating damage; and more, but we were able to solve all of these issues by working together as a team.

Overall, we are extremely proud of what we have accomplished through this process. All of us came in with little to no knowledge of Python and left with a respectable 2-player video game, along with a plethora of knowledge regarding Python. We discovered that working together was often the only way to accomplish difficult tasks while simpler tasks were most efficiently completed individually. This forced us to learn to communicate effectively, both verbally and through comments in the code, or we would waste precious time explaining our logic and process several times. We all agree that this experience has made each of us a more effective team member and enhanced our communication skills.