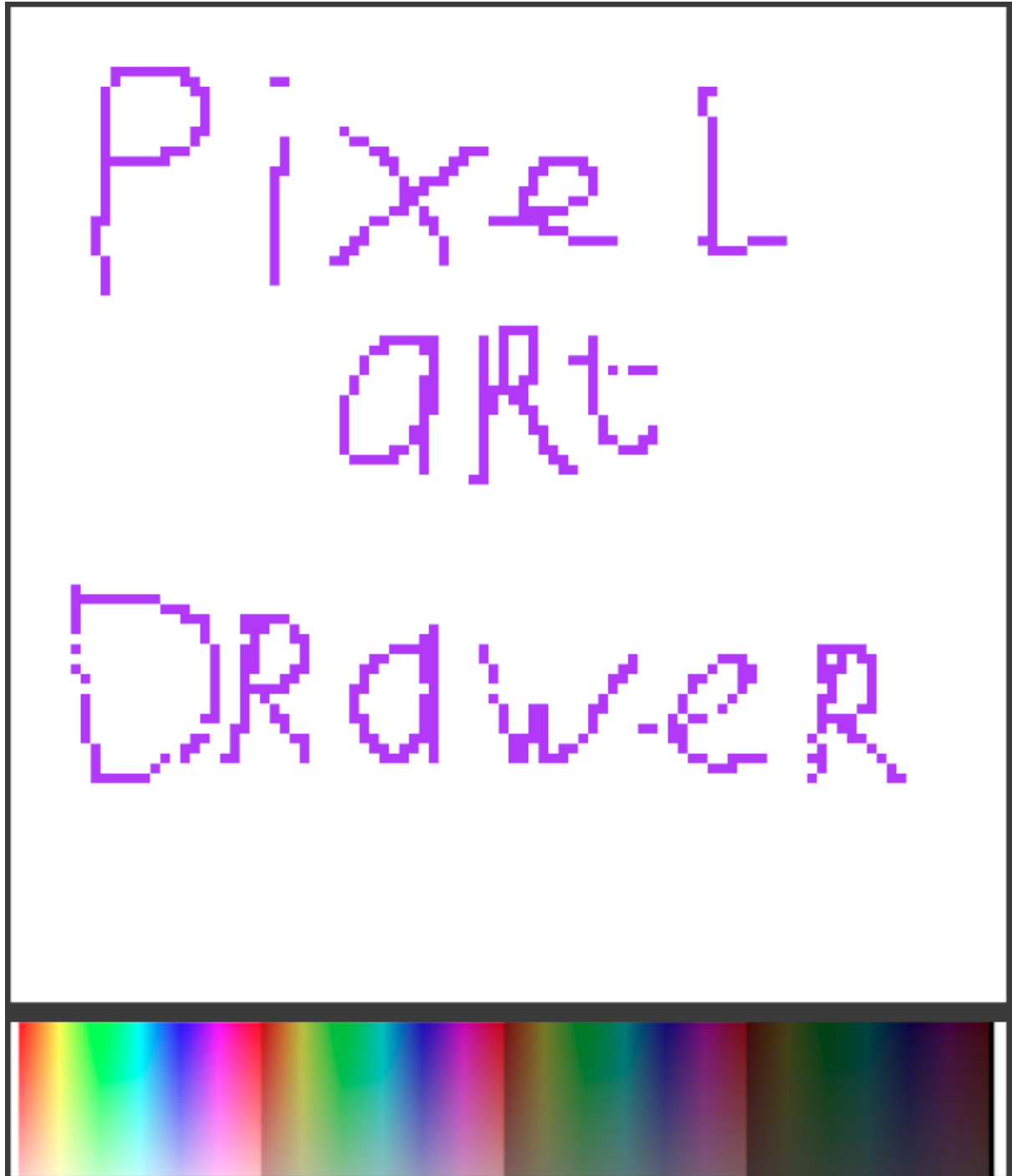


Pixel Art Drawing Program

Tool Concept

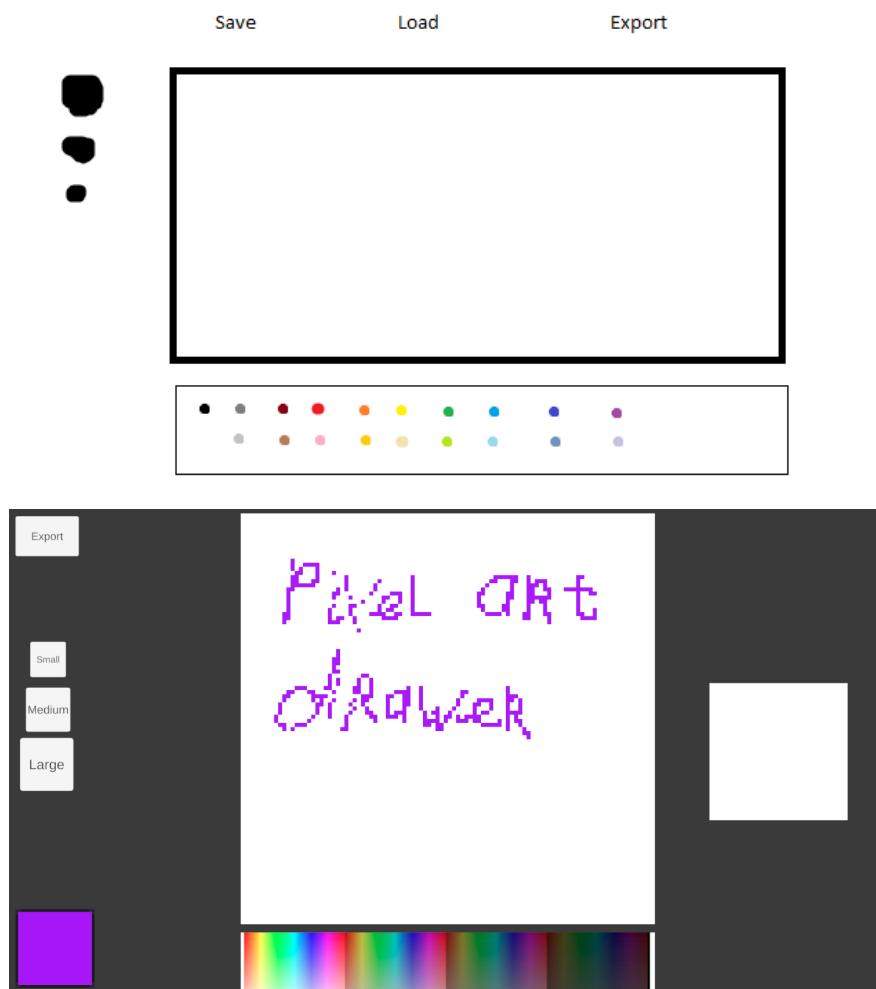


Door: Bradley van Ewijk, GI2 GDEV, 3048875

Concept

Voor deze opdracht ga ik een pixel art tekenprogramma maken met een sterke tutorial. Uit mijn gesprekken met game-artists kwamen twee dingen uit. Een was dat ze het vaak moeilijk vonden om een programma onder de knie te krijgen en het tweede was dat ze het leuk zouden vinden om een simpele tool te hebben voor snel pixelart maken. De meeste programma's zijn erg uitgebreid en ingewikkeld. Oorspronkelijk wilde ik een programma maken waarbij je meerdere programma's kon leren. Helaas paste dit idee niet helemaal bij de opdracht omtrent content creation. Na een tip van de docent heb ik besloten om de twee ideeën te combineren. Zo kan ik van beide werelden wat leren en in mijn opdracht stoppen.

Dus mijn concept is een simpel standalone pixelart tekenprogramma. Met niet te veel opties en onduidelijkheden. En in dit programma krijg je een canvasgrootte die je kan veranderen, een paar brush opties, een color picker, de standaard tools bij een tekenprogramma en een save, load en export optie. Voor het moment ga ik alleen exporteren naar .PNG, maar mocht er tijd zijn en het niet al te langdradig zijn zorg ik ook voor andere exportopties.

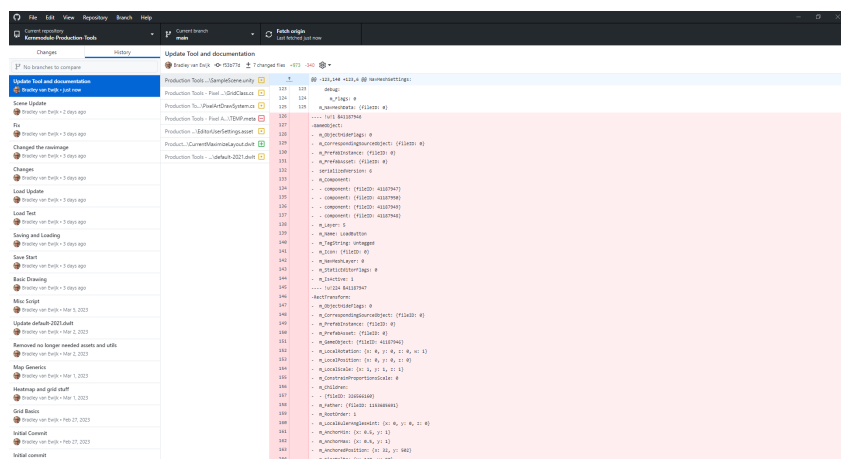
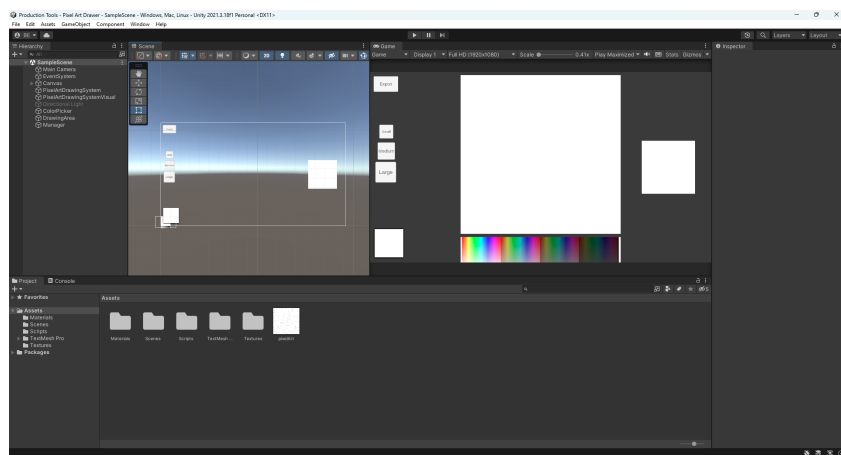


Process van Uitwerking

In het begin had ik geen idee waar ik moest starten en hoe ik het best kon beginnen. Zelf gebruik ik Aseprite voor pixelart. Dus ging ik eerst daar kijken naar de werking of ik wat kon vinden. Nu is Aseprite best ingewikkeld en zit er veel in dus werd mij dat snel te veel. Daarna ben ik wat gaan kijken op itch, maar ook dat werd snel overweldigend. Daarna ben ik maar wat gaan googlen. Daaruit kwamen wat ideeën, mijn favoriet was het gebruiken van een grid. Hier kon ik elke cell van het grid een kleur geven en zou je dat kunnen zien als pixels. Ook kan het grid groter en kleiner zijn als ik het goed doe. Ook zou ik dan verschillende brush sizes kunnen maken door het een range mee te geven.

Omdat we redelijk snel een prototype moeten hebben heb ik besloten het programma in Unity te maken. Hier heb ik nu zes jaar ervaring in en kan ik het snelst in werken. En natuurlijk staat alles op een Git repository. Omdat ik in mijn eentje werk, hou ik alles lekker op de main branch.

Mijn proces van uitwerking is niet echt veranderd en vooral hetzelfde gebleven. Ik ben wel iets meer gaan pushen. Na elke kleine verandering heb ik het op git gezet. Ook ben ik wat vaker gaan testen en builden. Zo kan ik er op tijd bij zijn als er wat mis gaat en kom ik daar niet pas aan het einde achter.



Technische onderbouwing

Op het moment van schrijven werkt er al redelijk wat. Het grid systeem is er, je kan een width, height, cellsize en de originposition instellen. Ook heb ik geëxperimenteerd met het tekenen door het maken van een heatmap. Ditzelfde principe kan ik straks toepassen tijdens het maken van de brushes, daar zit hetzelfde principe achter. Ook heb ik al nagedacht over eventuele toekomstige projecten en heb ik dingen aan het grid toegevoegd zodat ik het later voor nog meer dingen kan toepassen. Denk bijvoorbeeld aan grid based movement en neem als voorbeeld de Xcom games.

Ook heb ik een utility class gemaakt met handige functies die ik later kan gebruiken. Hier heb ik wel hulp gehad van iemand op youtube, namelijk code monkey. Zijn kanaal en links naar gebruikte video's zal ik in de links zetten.

Hierna ben ik verder gegaan met het tekenen zelf. Dit doe ik door de kleur te veranderen van een van mijn grid objecten. Ik werk niet met "echte" pixels, maar met grid objecten. Nadat ik het tekenen had toegevoegd kwamen er wat andere functies, met als eerste verschillende brush sizes. Dit doe ik met buttons, hij zoekt de knoppen op en voegt er een listener aan toe. En dan afhankelijk van op welke knop je drukt, zet hij met behulp van twee functies en een switch de brush groter of kleiner. Ik heb expres gekozen voor drie standaard groottes. Dit omdat het doel van mijn programma is om snel pixelart te maken en prototypen. Met een grotere instelbare brush wordt het al gauw te complex en verslaat dat de bedoeling van mijn programma.

Hierna heb ik een mogelijkheid toegevoegd om het gemaakte plaatje te exporteren naar een .png bestand. Dit doet hij door een 2D texture aan te maken, dan door het grid heen te lopen, die texture te applyen, dan om te zetten en encoden naar een png en dan naar een bestand te schrijven. Ook al had ik een versie van laden er alvast in zitten die het .png bestand op een raw image in de scene kan laden zodat je de afbeelding ook weer kan zien.

Het is me tot dusver nog niet gelukt om het ook weer editable te maken. Dit wordt de volgende stap. Dat wordt het grid serializable maken en alle data eruit opslaan naar een bestand. Dan dit weer uitlezen en dan weer editable maken, dit zou ik kunnen doen door mijn exportfunctie iets aan te passen. Hier ben ik met behulp van een docent achter gekomen.

Relevante UML

Vanwege drukte heb ik me gefocust op het maken van de tool en heb ik dit voor nu overgeslagen. Nogmaals mijn excuus voor dit, bij de tool deadline zal alles erin zitten. Ook pas ik nog heel wat dingen aan en maak ik daarom alles in 1 keer op het eind.

Links

Github link (public):

<https://github.com/PrimeLockerbee/Kernmodule-Production-Tools.git>

Code Monkey Channel:

<https://www.youtube.com/@CodeMonkeyUnity>

Grid System in Unity:

<https://www.youtube.com/watch?v=waEsGu--9P8&list=PLzDRvYVwl53uhO8yhqxcyjDImRjO9W722>

Powerful Generics Added Grid System in Unity:

<https://www.youtube.com/watch?v=8jrAWtl8RXg&list=PLzDRvYVwl53uhO8yhqxcyjDImRjO9W722&index=2>

Cool Heatmap in Unity:

<https://www.youtube.com/watch?v=mZzZXfySeFQ&list=PLzDRvYVwl53uhO8yhqxcyjDImRjO9W722&index=4>

Example code op Canvas:

<https://hku.instructure.com/courses/1592/files/folder/Code>