پروژه افزونه VS Code – WiseElder

معرفی پروژه:

هدف این پروژه، توسعه‌ی یک افزونه برای VS Code بود که یک شخصیت انیمیشنی تعاملی را در گوشه صفحه نمایش می‌دهد. این شخصیت به کاربران توصیه‌های سلامتی برای بهبود کیفیت جسمی و روحی در فواصل زمانی مشخص ارائه می‌کند و همچنین با کلیک کاربران، جملات حکیمانه‌ای ارائه می‌دهد.

مدت زمان و اعضای تیم:

این پروژه در مدت سه هفته توسط تیمی متشکل از **امیر شکیبافر، رضا روحانی، الهه آذرخشی** و **عرفان احراری** توسعه داده شد. نسخه اولیه آن پس از سه هفته در **مارکت VS Code** منتشر گردید.

مدیریت پروژه:

برای مدیریت و هماهنگی تسک‌ها از ابزار **Trello** استفاده شد. به دلیل محدودیت زمانی و پیچیدگی پروژه، تیم از متدولوژی **چابک (Agile)** بهره برد. جلسات هماهنگی هر سه شب یک‌بار برگزار می‌شد تا اهداف بعدی مشخص شوند و تطبیق مراحل پیشرفت پروژه با برنامه‌ریزی‌ها بررسی گردد.

اهداف اصلی:

مهم‌ترین هدف، طراحی و پیاده‌سازی انیمیشن‌های جذاب و کاربرپسند برای شخصیت انیمیشنی افزونه بود تا علاوه بر کاربردی بودن، تجربه بصری لذت‌بخشی به کاربران ارائه دهد.

تقسیم وظایف تیم:

۱. **امیر شکیبافر:**
   - مسئولیت هماهنگی جلسات و مدیریت کلی پروژه
   - پیاده‌سازی افزونه
۲. **رضا روحانی:**
   - کمک در پیاده سازی و استخراج نیازمندی‌ها
   - انتشار افزونه
   - مدیریت تسک‌ها در Trello
۳. **الهه آذرخشی:**
   - طراحی و پیاده‌سازی انیمیشن‌ها
   - تحلیل و استخراج نیازهای بصری
۴. **عرفان احراری:**
   - تحلیل نیازهای اولیه
   - نوشتن مستندات (صفحات README و نیازمندی‌ها)

فرآیند استخراج نیازمندی‌ها:

نیازمندی‌ها به فرمت **یوزر استوری (User Story)** تعریف شدند. هر یوزر استوری به تسک‌های جداگانه‌ای در Trello تبدیل شد که هر کدام اهداف مشخصی را دنبال می‌کردند.

نصب آسان

- **به‌عنوان** یک توسعه‌دهنده،
- **می‌خواهم** بتوانم افزونه را به سادگی از طریق فروشگاه افزونه‌های Visual Studio Code نصب کنم،
- **به‌طوری که** به راحتی آن را فعال کرده و شروع به استفاده کنم.

راه‌اندازی خودکار

- **به‌عنوان** یک کاربر،
- **می‌خواهم** پس از نصب، Wise Elderبه‌طور خودکار در نوار کناری ظاهر شود،
- **به‌طوری که** نیازی به پیکربندی اولیه نداشته باشم.

قابلیت‌های آفلاین

حالت آفلاین

- **به‌عنوان** یک کاربر،
- **می‌خواهم** Wise Elderحتی در حالت آفلاین بتواند نقل‌قول‌ها و یادآوری‌ها را ارائه دهد،
- **به‌طوری که** همیشه بتوانم از آن استفاده کنم.

استراحت چشم

- **به‌عنوان** یک توسعه‌دهنده،
- **می‌خواهم** هر 20 دقیقه یادآوری دریافت کنم تا استراحت کوتاهی به چشمانم بدهم،
- **به‌طوری که** از خستگی چشم ناشی از کار طولانی با صفحه‌نمایش جلوگیری کنم.

ایستادن و کشش

- **به‌عنوان** یک توسعه‌دهنده،
- **می‌خواهم** هر 90 دقیقه یادآوری دریافت کنم که بلند شوم و کمی بدنم را کشش دهم،
- **به‌طوری که** از خشکی عضلات و مشکلات ناشی از نشستن طولانی جلوگیری کنم.

نوشیدن آب

- **به‌عنوان** یک توسعه‌دهنده،
- **می‌خواهم** یادآوری منظم برای نوشیدن آب دریافت کنم،
- **به‌طوری که** هیدراته بمانم و انرژی بیشتری داشته باشم.

شخصیت متحرک

- **به‌عنوان** یک کاربر،
- **می‌خواهم** یک شخصیت انیمیشنی با ظاهر خردمندانه در نوار کناری ویرایشگر ببینم،
- **به‌طوری که** فضای کاری من جذاب‌تر و تعاملی‌تر شود.

نقل‌قول‌ها و جملات حکیمانه

نمایش نقل‌قول‌های تصادفی

- **به‌عنوان** یک کاربر،
- **می‌خواهم** با کلیک روی شخصیت Wise Elder یک جمله حکیمانه یا الهام‌بخش دریافت کنم،
- **به‌طوری که** در لحظات استراحت کمی الهام و انگیزه بگیرم.

کتابخانه نقل‌قول‌ها

- **به‌عنوان** یک کاربر،
- **می‌خواهم** نقل‌قول‌های متنوعی از موضوعات مختلف مانند الهام، بهره‌وری، و سلامت دریافت کنم،
- **به‌طوری که** تجربه کار با Wise Elder همیشه تازه و جذاب باشد.

معماری پروژه:
با توجه به بخش‌های متنوع و وظایف مجزای پروژه، طراحی معماری به صورت شی‌گرا انجام شد تا هر قسمت عملکرد خاص خود را داشته باشد و قابلیت توسعه و نگهداری بهتری فراهم شود. اجزای اصلی برنامه به صورت کلاس‌های مستقل و ماژولار پیاده‌سازی شدند، که هر کدام مسئولیت خاصی مانند مدیریت انیمیشن‌ها، تعامل با کاربر، زمان‌بندی پیام‌ها و مدیریت داده‌ها را بر عهده داشتند.

برای ایجاد انیمیشن‌ها و هماهنگ‌سازی آن‌ها با سایر بخش‌ها، تمامی کلاس‌ها در نهایت در یک اسکریپت اصلی مجتمع شدند. این اسکریپت درون یک **canvas** ادغام شد تا شخصیت انیمیشنی و سایر المان‌های گرافیکی به صورت یکپارچه و تعاملی در محیط افزونه نمایش داده شوند.

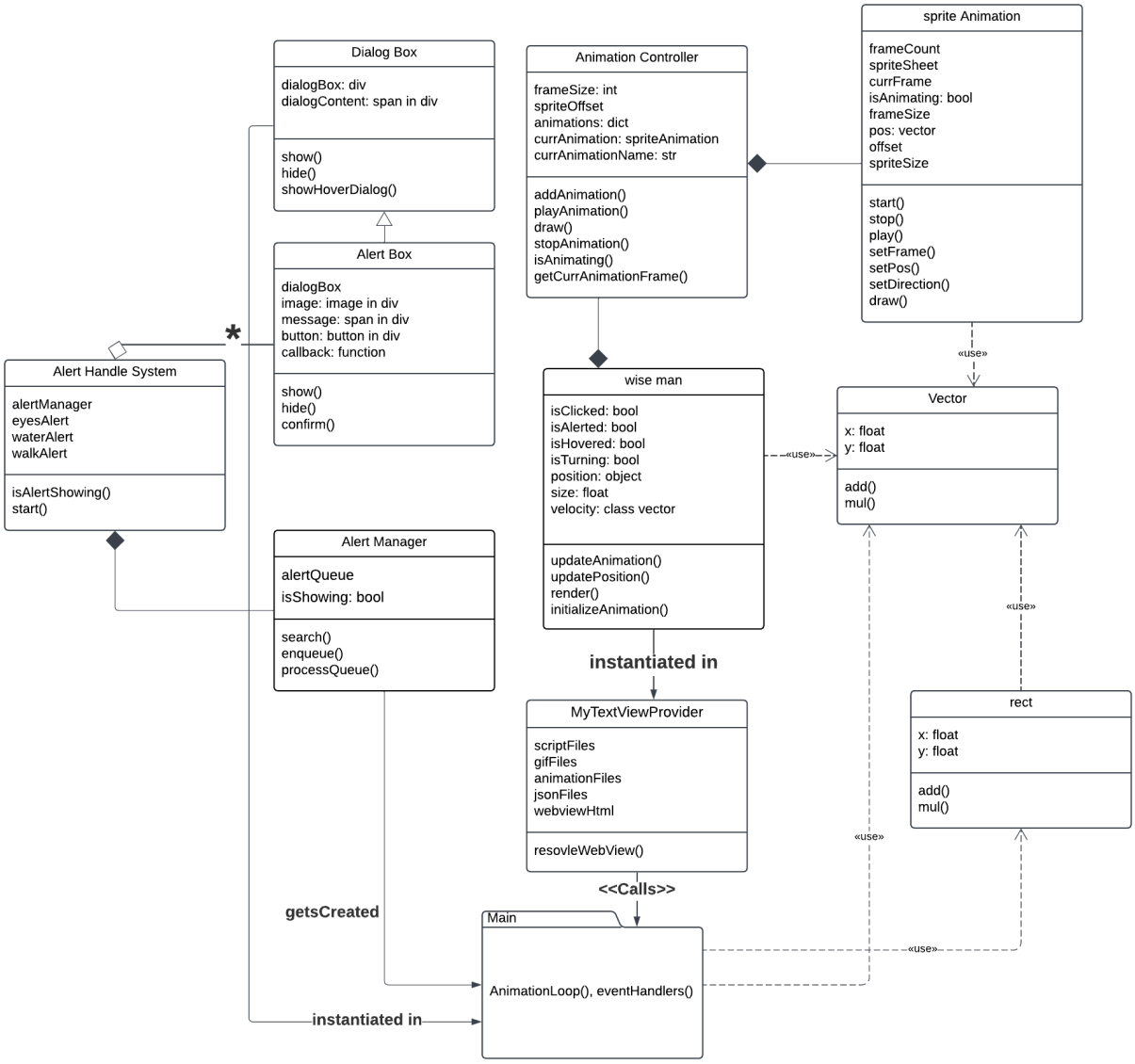
این ساختار، امکان انعطاف‌پذیری بالا را برای توسعه‌های آینده فراهم می‌کند و در عین حال، کدنویسی تمیز و منظم را تضمین می‌کند.

بهبودهای پیشنهادی در آینده:

- تفکیک بیشتر مسئولیت‌ها از طریق افزودن لایه‌های جداگانه برای مدیریت انیمیشن، منطق تجاری، و تعاملات کاربر.
- استفاده از الگوهای طراحی mvc برای افزایش قابلیت تست و مقیاس‌پذیری.
- اضافه کردن سیستم ماژولار برای بارگذاری کلاس‌ها به جای یکپارچه‌سازی آن‌ها در یک اسکریپت واحد.

این معماری فعلی با وجود سادگی، به نیازهای اصلی پروژه به خوبی پاسخ داده و تجربه کاربری روان و جذابی را ارائه می‌دهد.

شکل و ساختار کلی برنامه به کمک class diagram



پیاده‌سازی پروژه:

پیاده‌سازی پروژه افزونه **WiseMan** در چند مرحله اصلی انجام شد که هر کدام شامل فرآیندهای متعددی برای تضمین کیفیت، هماهنگی و دستیابی به اهداف نهایی پروژه بود. با توجه به ویژگی‌های متنوع این افزونه، تمام مراحل با دقت و برنامه‌ریزی دقیق پیش رفت.

مرحله اول: استخراج نیازمندی‌ها و برنامه‌ریزی اولیه

- **تحلیل نیازمندی‌ها :**نیازمندی‌ها در قالب **User Stories** استخراج شدند. این User Stories رفتارهای کاراکتر، تعاملات با کاربر، و زمان‌بندی پیام‌ها را تعریف کردند.
- **برنامه‌ریزی تسک‌ها :**نیازمندی‌ها به وظایف کوچک‌تر در ابزار **Trello** تبدیل شدند. این وظایف با توجه به اولویت‌ها و تخصص اعضای تیم تخصیص یافت.
- **هدف‌گذاری :**تمرکز اولیه بر ایجاد یک شخصیت انیمیشنی جذاب بود که بتواند تعاملات ساده اما مؤثری با کاربر برقرار کند.

مرحله دوم: طراحی و پیاده‌سازی اولیه

- **تعریف ساختار شی‌گرا:**
کلاس‌های اصلی پروژه شامل:
  ۱. **Wiseman class**مدیریت رفتارها و تعاملات کاراکتر.
  ۲. **Animation controller**: تولید و مدیریت انیمیشن‌ها.
  ۳. **Alert service handler**: زمان‌بندی پیام‌ها و هشدارها.
  ۴. **messagebox**: نمایش پیام‌ها و دریافت جملات حکیمانه از API.
  ۵. **CanvasRenderer**: ترکیب تمام عناصر گرافیکی و نمایش آن‌ها روی یک canvas.
- **ساختار اولیه: canvas**
یک محیط گرافیکی با استفاده از تگ canvas در محیط **VS Code WebView** ایجاد شد که کاراکتر انیمیشنی را نمایش می‌داد.
- **ایجاد انیمیشن‌ها:**
با استفاده از CSS و JavaScript ، حرکت‌های روان برای کاراکتر ایجاد شدند. حرکات شامل تغییر حالت‌ها (ایستادن، تکان دادن دست، چشمک زدن) بودند که با توجه به تعاملات کاربر تغییر می‌کردند.

مرحله سوم: یکپارچه‌سازی و تعامل با کاربر

- **توسعه API برای جملات حکیمانه:**
از یک API برای دریافت جملات حکیمانه و نمایش آن‌ها در پنجره هشدار استفاده شد.
- **ایجاد تایمر هشدارها:**
از setIntervalبرای زمان‌بندی هشدارها استفاده شد. هر ۲۰ دقیقه، کاراکتر پیام‌هایی مانند "چشم‌هایت را استراحت بده"، "یک لیوان آب بنوش" و "از پشت میز بلند شو" به کاربر نشان می‌داد.
- **تعامل‌پذیری:**
با کلیک روی کاراکتر، پنجره‌ای باز می‌شد که جمله‌ای حکیمانه یا توصیه‌ای سلامتی نمایش می‌داد.

مرحله چهارم: بهینه‌سازی و انتشار

- **بررسی عملکرد:**
افزونه در شرایط مختلف آزمایش شد تا از روان بودن انیمیشن‌ها و عملکرد صحیح پیام‌ها اطمینان حاصل شود.
- **بهینه‌سازی کد:**
کدهای پروژه با تمرکز بر کاهش وابستگی‌ها و افزایش خوانایی بازنویسی شدند.
- **تولید مستندات:**
فایل‌های README شامل توضیحات افزونه، نحوه استفاده، و ویژگی‌های اصلی تکمیل شدند.
- **انتشار:**
افزونه پس از تکمیل در **مارکت VS Code** بارگذاری شد و نسخه اولیه در دسترس کاربران قرار گرفت.

روند کلی توسعه:

- **هماهنگی تیمی :**جلسات منظم هر سه شب یک‌بار برگزار می‌شد تا وظایف انجام‌شده بررسی و اهداف جدید تعیین شوند.
- **انعطاف‌پذیری متدولوژی چابک :**تغییرات در اولویت‌ها یا ویژگی‌ها به سرعت اعمال شدند تا با محدودیت زمانی سه هفته‌ای سازگار شویم.

چالش‌های پروژه:

1. **هماهنگی انیمیشن‌ها با:canvas**
از جمله چالش‌های فنی اصلی، هماهنگی انیمیشن‌ها با محیط canvas و اجرای روان آن‌ها بود. این چالش به دلیل پیچیدگی‌های فنی و ناآشنایی اولیه تیم با ابزارهای مرتبط ایجاد شد، اما با تلاش تیمی و یادگیری مستمر برطرف شد.
2. **تمرکز ناکافی در استخراج نیازمندی‌ها:**
عدم تمرکز کافی در فاز نیازمندی‌ها منجر به تعریف ناقص یا مبهم برخی اهداف شد. این موضوع در نهایت باعث کاهش کارایی و کیفیت کلی سیستم گردید، زیرا تسک‌ها گاهی اوقات بدون درک کامل اولویت‌ها و نیازها اجرا شدند.
3. **وعده‌های بیش از حد در ابتدای پروژه:**
ضعف مدیریتی در شروع پروژه منجر به دادن قول‌های غیرواقعی شد. این قول‌ها فشار زیادی بر تیم وارد کرد و نتوانستیم برخی ویژگی‌ها را با کیفیتی مطلوب تحویل دهیم.
4. **برنامه‌ریزی ناقص برای پیاده‌سازی و تست هشدارها:**
عدم وجود برنامه دقیق برای توسعه و تست هشدارها منجر به کشف مشکلات در اواخر پروژه شد. زمان باقی‌مانده برای رفع این اشکالات کافی نبود و برخی از آن‌ها به نسخه اولیه افزونه منتقل شدند.
5. **تخصیص نامناسب فعالیت‌ها:**
تخصیص نادرست وظایف باعث شد برخی از اعضا نتوانند توانایی‌های خود را به طور کامل نشان دهند. به‌عنوان مثال، آقای احراری با اینکه وظایف خود را انجام دادند، شاید اگر در جایگاه بهتری قرار می‌گرفتند، مشارکت بیشتری نشان می‌دادند و خروجی بهتری ارائه می‌کردند.
6. **تمرکز کدنویسی تنها روی دو نفر:**
تصمیم برای محدود کردن کدنویسی به دو نفر باعث شد مشکلات موجود در روند پیاده‌سازی هشدارها دیر کشف شود و فرصتی برای رفع آن‌ها در نسخه اولیه باقی نماند.
7. **عدم تجربه تیم در معماری و طراحی اولیه:**
به دلیل محدودیت دانش فنی کل تیم و زمان‌بر بودن یادگیری، امکان ایجاد یک برنامه جامع برای معماری وجود نداشت. این موضوع منجر به چندین بار **refactor**در کدها شد که زمان و انرژی تیم را کاهش داد.

8. **ضعف مدیریتی:**
تجربه کم در مدیریت تیم باعث شد از ابتدا ویژگی‌های پیچیده‌ای به پروژه اضافه شود که پیاده‌سازی آن‌ها چالش‌برانگیز بود. همچنین، برنامه‌ریزی ناقص برای تست اصولی و تخصیص نامناسب وظایف منجر به کاهش بهره‌وری و کشف مشکلات در اواخر پروژه گردید.

نکات مثبت و تلاش‌های تیم:
با وجود تمام چالش‌ها و مشکلات، تلاش‌های تیم باعث شد نسخه اولیه پروژه به نتیجه برسد:

- **الهه آذرخشی** بدون هیچ تجربه قبلی در انیمیشن‌سازی، شخصیتی بامزه و دوست‌داشتنی طراحی و پیاده‌سازی کرد که نقطه قوت بصری پروژه شد.
- **رضا روحانی** در تمامی بن‌بست‌ها و چالش‌ها با ارائه راه‌حل‌های خلاقانه نقش بسیار موثری داشت و در مواردی که مدیریت پروژه دچار مشکل می‌شد، به عنوان پشتیبان کمک بزرگی بود.
- **عرفان احراری** وظایف خود را تحویل دادند، اما اگر پیگیری بیشتری از سوی مدیریت انجام می‌شد، امکان بهبود کیفیت خروجی ایشان نیز فراهم می‌شد.

درس‌هایی برای آینده:

1. **تمرکز بیشتر بر استخراج نیازمندی‌ها :**تعریف دقیق و اولویت‌بندی مناسب اهداف پروژه می‌تواند درک روشنی از وظایف ایجاد کند.
2. **بهبود مدیریت وظایف و ارتباطات :**تخصیص بهتر کارها و بهره‌گیری از توانایی‌های تمامی اعضای تیم باعث افزایش بهره‌وری می‌شود.
3. **ایجاد برنامه دقیق برای تست :**شناسایی مشکلات در مراحل اولیه توسعه به صرفه‌جویی در زمان و افزایش کیفیت منجر خواهد شد.
4. **پرهیز از قول‌های غیرواقعی :**هدف‌گذاری منطقی و هم‌سو با توانایی‌ها و زمان‌بندی تیم به کاهش فشار و بهبود خروجی کمک می‌کند.

در نهایت، این پروژه تجربه‌ای ارزشمند برای تیم و شخص مدیر پروژه بود که زمینه‌ساز یادگیری عمیق‌تری در زمینه توسعه و مدیریت پروژه‌های نرم‌افزاری شد.